A
Data science
Project
on

“Ratings Prediction”

Submitted by:

Santosh Arvind Dharam

ACKNOWLEDGMENT

I feel great pleasure to present the Project entitled “Ratings Prediction”. But it would be unfair on our part if I do not acknowledge efforts of some of the people without the support of whom, this Project would not have been a success. First and for most I am very much thankful to my respected SME ‘Swati Mahaseth’ for his leading guidance in this Project. Also he has been persistent source of inspiration to me. I would like to express my sincere thanks and appreciation to ‘flip robo’ for their valuable support. Most importantly I would like to express our sincere gratitude towards my Friend & Family for always being there when I needed them most.

Mr. Santosh Arvind Dharam

INTRODUCTION

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 i.e. the reviewer will have to add stars (rating) as well with the review. 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

PROBLEM STATEMENT

We have to build an application which can predict the rating by seeing the review.

Analytical Problem Framing

EDA steps:

1) import necessary libraries:

first we will import all the necessary libraries which will be usefull for analysis of data

```
In [1]: #import all libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
from sklearn.linear_model import LogisticRegression,Lasso,LinearRegression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import AdaBoostRegressor,GradientBoostingRegressor
from sklearn.preprocessing import LabelEncoder,StandardScaler
from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.decomposition import PCA
from scipy.stats import zscore
from sklearn.model_selection import cross_val_score
```

in this case we have to import all the necessary library that are useful for data analysis in jupyter notebook

2) Extract the dataset in jupyter notebook:

```
In [2]: data=pd.read_excel("C:\\Users\\SAI BABA\\Desktop\\amazonReview1.xlsx")
data.head()
```

Out[2]:

	Reviewer name	reviews of the product	Review body	rating of the product	sentiment
0	Genuine buyer	Pros and cons	Great quality and the sound is actually great....	5.0 out of 5 stars	positive
1	CHETAN SHETTER	Never Expected in this price range I Assure Yo...	I've been using this from past 6 months. This ...	5.0 out of 5 stars	positive
2	Kiran Kumar M	Bass bastler, Awesome HD quality sound, rocking	Perfect bass and nice one. Superb sound and it...	5.0 out of 5 stars	positive
3	Raj Patel	Maybe buy boat basshead	Here is the review after 1 week of usePros:1.B...	4.0 out of 5 stars	positive
4	SANJAY KUMAR TIWARI	Superb head stereo better than leading brands.	Purchased after seeing review with suspicion t...	5.0 out of 5 stars	positive

```
In [3]: data.shape
```

Out[3]: (449779, 5)

Data is extracted for further analysis in jupyter notebook data contains a 449779 rows and 5 columns some columns contains object type data and some contains numerical.

3) checking null values:

In this case we have to find out the null values present in our data set. if it is there it is required to remove it. in our data set has some null values which is shown below

```
In [5]: data.isnull().sum()

Out[5]: Reviewer name                13
        reviews of the product       61
        Review body                 157
        rating of the product     36406
        sentiment                     0
        dtype: int64
```

4) Removing Null Values:

```
[6]: data['Reviewer name']=data['Reviewer name'].fillna(data['Reviewer name'].mode()[0])
     data['reviews of the product']=data['reviews of the product'].fillna(data['reviews of the product'].mode()[0])
     data['Review body']=data['Review body'].fillna(data['Review body'].mode()[0])
     data['rating of the product']=data['rating of the product'].fillna(data['rating of the product'].mode()[0])
```

Null values which are present in the dataset is removed by the mode function

```
In [7]: data.isnull().sum()

Out[7]: Reviewer name                0
        reviews of the product       0
        Review body                  0
        rating of the product        0
        sentiment                    0
        dtype: int64
```

5)Data Encoding:

It is necessary to encode the data as the data contains object type of data , so it is need to convert into the integer type for the further analysis. it is possible with the help of labelEncoder

```
In [9]: from sklearn.preprocessing import LabelEncoder
```

```
In [10]: le=LabelEncoder()
label=le.fit_transform(data["rating of the product"])
label
data=data.drop("rating of the product",axis='columns')
data["rating of the product"]=label
```

```
In [11]: data
```

Out[11]:

	Reviewer name	reviews of the product	Review body	sentiment	rating of the product
0	Genuine buyer	Pros and cons	Great quality and the sound is actually great....	positive	4
1	CHETAN SHETTER	Never Expected in this price range I Assure Yo...	I've been using this from past 6 months. This ...	positive	4
2	Kiran Kumar M	Bass bastler, Awesome HD quality sound, rocking	Perfect bass and nice one. Superb sound and it...	positive	4
3	Raj Patel	Maybe buy boat basshead	Here is the review after 1 week of usePros:1.B...	positive	3
4	SANJAY KUMAR TIWARI	Superb head stereo better than leading brands.	Purchased after seeing review with suspicion t...	positive	4
...	...	...	...	...	...
449774	Dipchand	Not good	One side head phone is not working within 10 d...	negative	0
449775	Amazon Customer	Need replacement	Quality is good. And there is some damage near...	negative	0
449776	MAYANK MISHRA	Stopped working after 1 month.	Your browser does not support HTML5 video. Â A...	negative	2
449777	Zan	Cheap Quality.	This product is of very cheap quality. Wouldn'...	negative	0
449778	SK Raju	First few days only it works	Bad luck to buy, within month went bad does no...	negative	1

Also it is need to drop some columns as it has no use so will also drop that columns

In [12]:
```
data.drop(['Reviewer name'],axis=1,inplace=True)
```

In [13]:
```
data
```

Out[13]:

	reviews of the product	Review body	sentiment	rating of the product
0	Pros and cons	Great quality and the sound is actually great....	positive	4
1	Never Expected in this price range I Assure Yo...	I've been using this from past 6 months. This ...	positive	4
2	Bass bastler, Awesome HD quality sound, rocking	Perfect bass and nice one. Superb sound and it...	positive	4
3	Maybe buy boat basshead	Here is the review after 1 week of usePros:1.B...	positive	3
4	Superb head stereo better than leading brands.	Purchased after seeing review with suspicion t...	positive	4
...	...	...	...	...
449774	Not good	One side head phone is not working within 10 d...	negative	0
449775	Need replacement	Quality is good. And there is some damage near...	negative	0
449776	Stopped working after 1 month.	Your browser does not support HTML5 video. Â A...	negative	2
449777	Cheap Quality.	This product is of very cheap quality. Wouldn'...	negative	0
449778	First few days only it works	Bad luck to buy, within month went bad does no...	negative	1

In [14]:
```
le=LabelEncoder()
label=le.fit_transform(data["sentiment"])
label
data=data.drop("sentiment",axis='columns')
data["sentiment"]=label
data
```

In [15]:
```
data.drop(['Review body'],axis=1,inplace=True)
```

In [16]:
```
data
```

Out[16]:

	reviews of the product	rating of the product	sentiment
0	Pros and cons	4	1
1	Never Expected in this price range I Assure Yo...	4	1
2	Bass bastler, Awesome HD quality sound, rocking	4	1
3	Maybe buy boat basshead	3	1
4	Superb head stereo better than leading brands.	4	1
...	...	...	...
449774	Not good	0	0
449775	Need replacement	0	0
449776	Stopped working after 1 month.	2	0
449777	Cheap Quality.	0	0
449778	First few days only it works	1	0

449779 rows × 3 columns

```
In [18]: # Convert all messages to lower case
data['reviews of the product'] = data['reviews of the product'].str.lower()

# Replace email addresses with 'email'
data['reviews of the product'] = data['reviews of the product'].str.replace(r'^.+@[^\.].*\.[a-z]{2,}$',
                                 'emailaddress')

# Replace URLs with 'webaddress'
data['reviews of the product'] = data['reviews of the product'].str.replace(r'^http\://[a-zA-Z0-9\-\.]+\.[a-zA-Z]{2,3}(/\S*)?$',
                                 'webaddress')

# Replace money symbols with 'moneysymb' (£ can by typed with ALT key + 156)
data['reviews of the product'] = data['reviews of the product'].str.replace(r'£|\$', 'dollers')

# Replace 10 digit phone numbers (formats include paranthesis, spaces, no spaces, dashes) with 'phonenumber'
data['reviews of the product'] = data['reviews of the product'].str.replace(r'^\(?[\d]{3}\)?[\s-]?[\d]{3}[\s-]?[\d]{4}$',
                                 'phonenumber')


# Replace numbers with 'numbr'
data['reviews of the product'] = data['reviews of the product'].str.replace(r'\d+(\.\d+)?', 'numbr')
```

```
In [19]: le=LabelEncoder()
label=le.fit_transform(data["reviews of the product"])
label
data=data.drop("reviews of the product",axis='columns')
data["reviews of the product"]=label
data
```

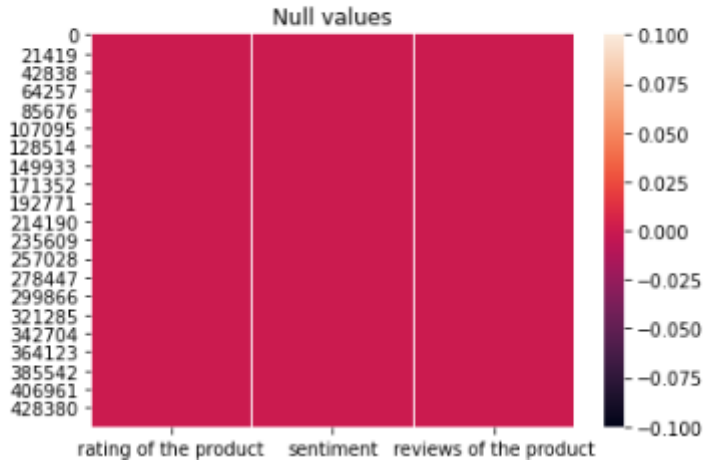

Also in the column of ‘reviews of the product’ it contains data in mix form so it is need to convert into the integer form that can be done by doing some steps as above.

6)heat map:

```
In [21]: data.isnull().sum()
```

```
Out[21]: rating of the product     0
         sentiment                 0
         reviews of the product    0
         dtype: int64
```

```
In [22]: sns.heatmap(data.isnull())
         plt.title("Null values")
         plt.show()
```



So all the null values are removed now ,it is also shown by the heat map as above

7)Data Description:

```
In [23]: data.describe()
```

Out[23]:

	rating of the product	sentiment	reviews of the product
count	449779.000000	449779.000000	449779.000000
mean	3.290294	0.851801	88234.401288
std	1.177584	0.355298	50706.045663
min	0.000000	0.000000	0.000000
25%	3.000000	1.000000	50934.000000
50%	4.000000	1.000000	80997.000000
75%	4.000000	1.000000	131478.500000
max	4.000000	1.000000	178494.000000

It gives the detail description of data with total counts ,mean,with its std deviation.also it gives mini to maximum values present in that particular column.as the value of std deviation is less as compare to mean ,it shows that our data is well for further analysis.

8)Data correlation:

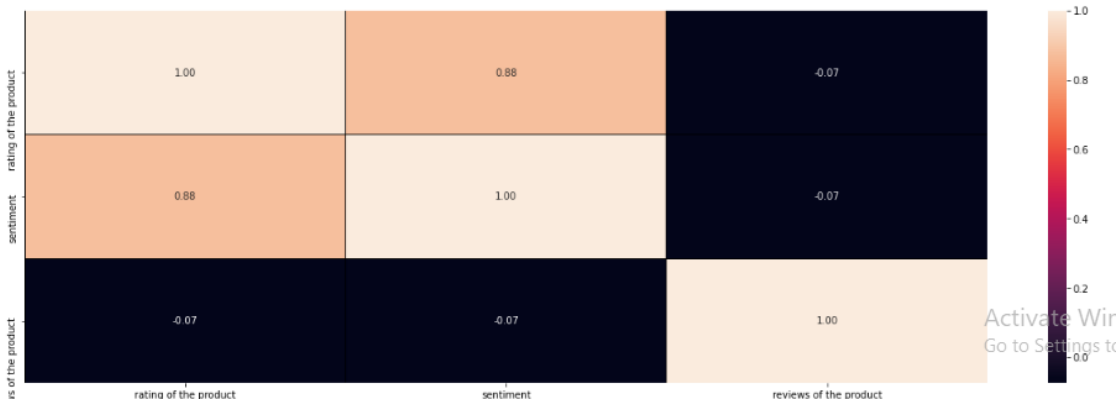
```
In [24]: data.corr()
```

Out[24]:

	rating of the product	sentiment	reviews of the product
rating of the product	1.000000	0.877381	-0.074488
sentiment	0.877381	1.000000	-0.070817
reviews of the product	-0.074488	-0.070817	1.000000

```
In [25]: #heat map
plt.figure(figsize=(22,7))
sns.heatmap(data.corr(),annot=True,linewidths=0.1,linecolor='black',fmt='0.2f')
```

Out[25]: <AxesSubplot:>



It gives the correlation of targe variable with the other column ,it also gives positive negative correlation of each column .

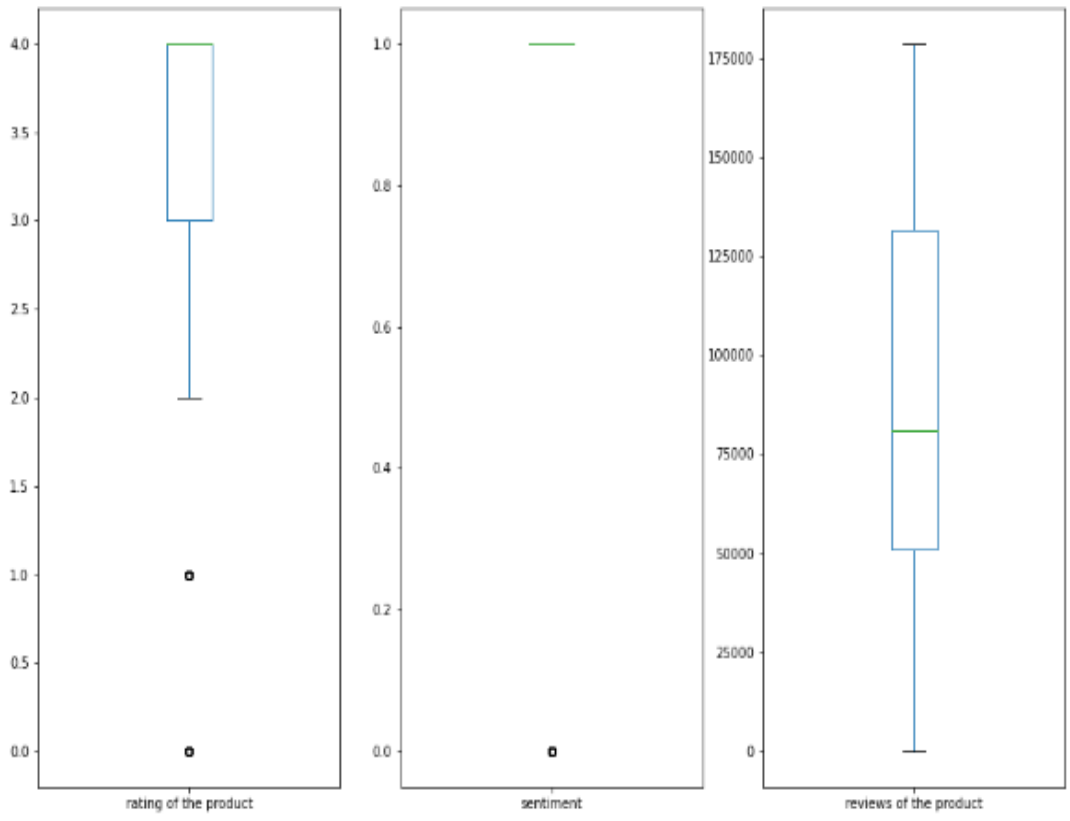
9)Finding and Removing the Skewness :

```
In [26]: # checking the skewness for the features:
         data.skew()

Out[26]: rating of the product    -1.831271
         sentiment                -1.980322
         reviews of the product    0.116532
         dtype: float64
```

it shows that there is no skewness present in our dataset

10) checking the outliers:

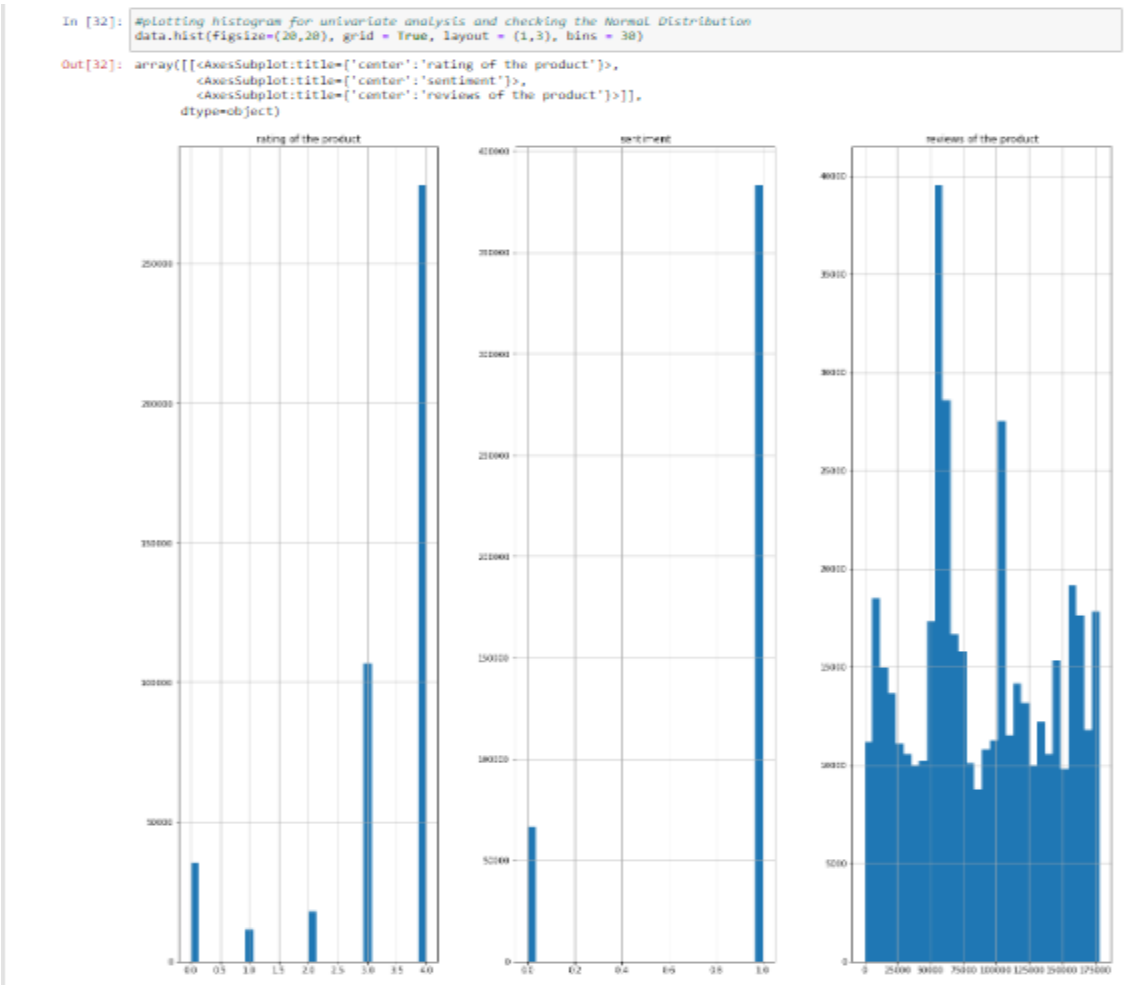
```
In [28]: #plotting the boxplot of each column to check the outliers
         data.plot(kind='box',subplots = True,layout=(1,3),figsize = (15,10))

Out[28]: rating of the product        AxesSubplot(0.125,0.125;0.227941x0.755)
         sentiment                AxesSubplot(0.398529,0.125;0.227941x0.755)
         reviews of the product   AxesSubplot(0.672059,0.125;0.227941x0.755)
         dtype: object
```



It shows that there is no outliers present in dataset.

11)plotting of histogram:

```
In [32]: #plotting histogram for univariate analysis and checking the Normal Distribution
         data.hist(figsize=(20,20), grid = True, layout = (1,3), bins = 30)

Out[32]: array([[<AxesSubplot:title={'center':'rating of the product'}>,
                 <AxesSubplot:title={'center':'sentiment'}>,
                 <AxesSubplot:title={'center':'reviews of the product'}>]],
               dtype=object)
```



It shows that data is uniformely distributed in each column

12) Divide data into x and y variable:

Now let's we will divide data into two variable x and y for further analysis

```
In [30]: #assign the value of x and y for training and testing phase
         x = data.drop(columns=['rating of the product'])
         y = data[["rating of the product"]]
         print(x.shape)
         print(y.shape)
```

```
(449779, 2)
(449779, 1)
```

Now lets we use multiple algorithm and will select best among them

13) using multiple Algorithms:

```
In [31]: #Standardize the value of x so that mean will 0 and SD will become 1 , and make the data as normal distributed
sc = StandardScaler()
sc.fit_transform(x)
x = pd.DataFrame(x,columns=x.columns)
```

```
In [*]: #Now by using multiple Algorithms we are calculating the best Algo which suit best for our data set

model = [DecisionTreeRegressor(),KNeighborsRegressor(),AdaBoostRegressor(),LinearRegression(),GradientBoostingRegressor()]
max_r2_score = 0
for r_state in range(1,50):
    train_x,test_x,train_y,test_y = train_test_split(x,y,random_state = r_state,test_size = 0.24)
    for i in model:
        i.fit(train_x,train_y)
        pre = i.predict(test_x)
        r2_sc = r2_score(test_y,pre)
        print("R2 score correspond to random state " ,r_state ,"is", r2_sc)
        if r2_sc> max_r2_score:
            max_r2_score=r2_sc
            final_state = r_state
            final_model = i

print()
print()
print()
print()
print("max R2 score correspond to random state " ,final_state , "is" , max_r2_score ,"and model is",final_model)
```

```
R2 score correspond to random state  1 is 0.7459957705355454
R2 score correspond to random state  1 is 0.5598293760838253
R2 score correspond to random state  1 is 0.7415233785597602
R2 score correspond to random state  1 is 0.7682332981886129
R2 score correspond to random state  1 is 0.7890053764958606
R2 score correspond to random state  2 is 0.7490767048414311
R2 score correspond to random state  2 is 0.5571270233152144
R2 score correspond to random state  2 is 0.7449502565091366
R2 score correspond to random state  2 is 0.7700078641642641
R2 score correspond to random state  2 is 0.7898222786625464
R2 score correspond to random state  3 is 0.7497766070649525
R2 score correspond to random state  3 is 0.5653350025200762
R2 score correspond to random state  3 is 0.7378306496300502
R2 score correspond to random state  3 is 0.7700552820114656
```

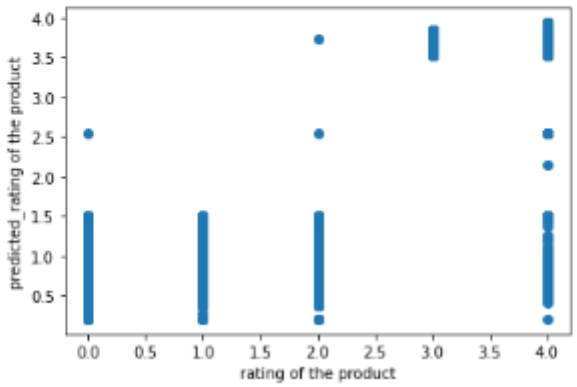
```
R2 score correspond to random state  49 is 0.749573022571603
R2 score correspond to random state  49 is 0.5618678973655603
R2 score correspond to random state  49 is 0.7413115874625463
R2 score correspond to random state  49 is 0.7701819967404006
R2 score correspond to random state  49 is 0.790479751381479
```

```
max R2 score correspond to random state  44 is 0.7934816995515146 and model is GradientBoostingRegressor()
```

So we got maximum R2score for random state of 44 is 79.34% foe model of GradientBoostingRegressor()

14)Scatter Plot:

```
In [34]: #checking the diff between actual and predicted value using graph
         plt.scatter(x=test_y,y=pred)
         plt.xlabel('rating of the product')
         plt.ylabel('predicted_rating of the product')

Out[34]: Text(0, 0.5, 'predicted_rating of the product')
```



It gives the actual vs predicted rating of product ,they are very close to each other

15)Saving Model:

```
In [35]: import pickle

In [36]: #saving model to the local file system
         filename='Ratings Prediction.pickle'
         pickle.dump(gbr,open(filename,'wb'))

In [37]: filename

Out[37]: 'Ratings Prediction.pickle'
```

We have saved the model with the file name as ratings prediction.

16)Hyper parameter Tuning:

```
In [41]: from sklearn.model_selection import GridSearchCV
         from sklearn.model_selection import cross_val_score
         import warnings
         from sklearn.linear_model import Lasso
         warnings.filterwarnings('ignore')

In [42]: parameters={'alpha':[.0001,0.001,.01,.1,1,10],'random_state':list(range(0,30))}
         ls=Lasso()
         clf=GridSearchCV(ls,parameters)
         clf.fit(x_train,y_train)
         print(clf.best_params_)

         {'alpha': 0.0001, 'random_state': 0}

In [43]: from sklearn.metrics import r2_score

In [44]: ls=Lasso(alpha=0.0001,random_state=0)
         ls.fit(x_train,y_train)
         ls.score(x_train,y_train)
         pred_ls=ls.predict(x_test)
         lss=r2_score(y_test,pred_ls)
         lss

Out[44]: 0.7730117946969384
```

So by doing the hyper parameter tuning we got alpha as 0.0001 with random state of 0, from this we got r2 score as 77.30%.

17)conclusion:

```
In [39]: x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.24,random_state=44)

In [40]: #Lets predict data
         y_pred=gbr.predict(x_test)
         y_pred

Out[40]: array([0.98089807, 0.98089807, 0.98089807, ..., 0.98089807, 0.98089807,
                0.98089807])
```

We observed that data was not filled with outliers also there some encoding is done now data is uniformly distributed in column. it is also observed from subplot, we have saved model and also predicted the result with help of saved model .model is ready for the future data prediction.