

RATING PREDICTION

Submitted by:

Amritesh Kumar.

# Acknowledgement

I would like to express my gratitude to my guide Shubham Yadav (SME, Flip Robo) for his constant guidance, continuous encouragement and unconditional help towards the development of the project. It was he who helped me whenever I got stuck somewhere in between. The project would have not been completed without his support and confidence he showed towards me.

Lastly, I would l like to thank all those who helped me directly or indirectly toward the successful completion of the project.

# Introduction

## Business Problem Framing

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. So, we have to build an application which can predict the rating by seeing the review. As a Data Scientist, we have to apply our analytical skills to give findings and conclusions in detailed data analysis written in Jupyter notebook.

## Conceptual Background of the Domain Problem

Created model can be used to predict ratings of reviews, it might be a good tool for online shopping sites such as Flipkart.com, Amazon.in etc and manufacturer companies who might look into their product ratings and reviews so that they can make their investment according to demand of customers , which might help them to save time and earn more profits.

## Review of Literature

Now a days people buy more products from online sites, So by using this model, rating prediction and positive or negative review prediction of a product is easy and less time consuming.

## Motivation for the Problem Undertaken

Data Science help us to make predictions at areas like health sectors, auto industry, education, media etc. For our project we decided to implement Prediction of Rating of reviews. We have to create a model which will help online sites and manufacturer of a product if they have to modify their product or not according to customers requirement

# Analytical Problem Framing

## Mathematical/ Analytical Modelling of the Problem

We would perform one type of supervised learning algorithms: classification. While it seems more reasonable to perform classification since we have 5 types of Rating i.e., 1, 2, 3, 4, 5. The prediction will base on the dataset which is scrapped from Flipkart and Amazon technical product. Scrapping dataset contains reviews and ratings of different products.

## Data Sources and their formats

**We scraped more than 20000 rows of data. We can scrape more data as well. More the data better the model. In this section we need to scrape the reviews of different laptops, Phones, Headphones, smart watches, Professional Cameras, Printers, Monitors, Home theatre, Router from different e-commerce websites. Basically, we need these columns**

**1) reviews of the product.**

**2) rating of the product. We fetched other data as well. Our columns are –**

**1. Rating**

**2. Review**

**3. Long Review**

****

## Data Pre-processing Done

Cleaned the data from junk values. Replace multiple spaces with single space So that it’ll easy to classify it.

## Hardware and Software Requirements and Tools Used

The system requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client’s point of view. System requirements are all of the requirements at the system level that describe the functions which the system as a whole should fulfil to satisfy the stakeholder needs and requirements, and is expressed in an appropriate combination of textual statements, views, and non-functional requirements; the latter expressing the levels of safety, security, reliability, etc., that will be necessary.

**Hardware requirements**: -

1. Processor — core i5 and above

2. RAM — 8 GB or above

3. SSD — 250GB or above

**Software requirements**: -

Anaconda

**Libraries**: -

**From sklearn.preprocessing import StandardScaler**

As these columns are different in **scale**, they are **standardized** to have common **scale** while building machine learning model. This is useful when you want to compare data that correspond to different units.

**from sklearn.preprocessing import Label Encoder**

 Label Encoder  and One Hot Encoder. These two encoders are parts of the SciKit Learn library in Python, and they are used to convert categorical data, or text data, into numbers, which our predictive models can better understand.

**From sklearn.model\_selection import train\_test\_split,cross\_val\_score**

Train\_test\_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train\_test\_split will make random partitions for the two subsets.

The algorithm is trained and tested K times, each time a new set is used as testing set while remaining sets are used for training. Finally, the result of the K-Fold Cross-Validation is the average of the results obtained on each set.

**from sklearn.neighbors import KNeighborsClassifier**

K Nearest Neighbor(KNN) is a very simple, easy to understand, versatile and one of the topmost machine learning algorithms. KNN used in the variety of applications such as finance, healthcare, political science, handwriting detection, image recognition and video recognition

**from sklearn.linear\_model import LogisticRegression**

The library sklearn can be used to perform logistic regression in a few lines as shown using the LogisticRegression class. It also supports multiple features. It requires the input values to be in a specific format hence they have been reshaped before training using the fit method.

**from sklearn.tree import DecisionTreeClassifier**

Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and number of attributes in the given data. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability distribution assumptions. Decision trees can handle high dimensional data with good accuracy.

# Model/s Development and Evaluation

## Identification of possible problem-solving approaches (methods)

Just make the comments more appropriate so that we’ll get less word to process and get more accuracy. Removed extra spaces, converted email address into email keyword, likely wise phone number etc. Tried to make Comments small and more appropriate as much as it was possible.

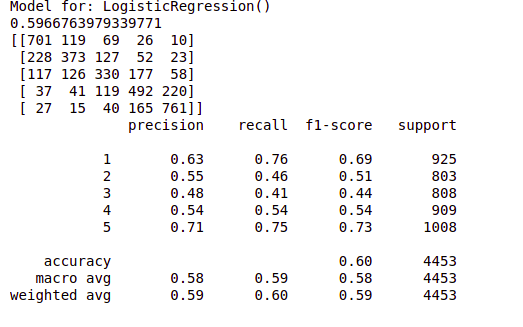
## Testing of Identified Approaches (Algorithms)

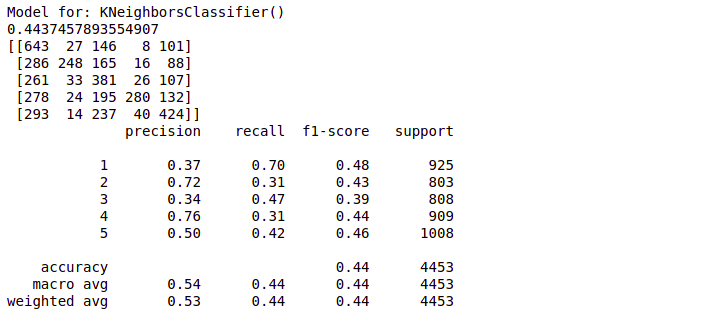
Listing down all the algorithms used for the training and testing.

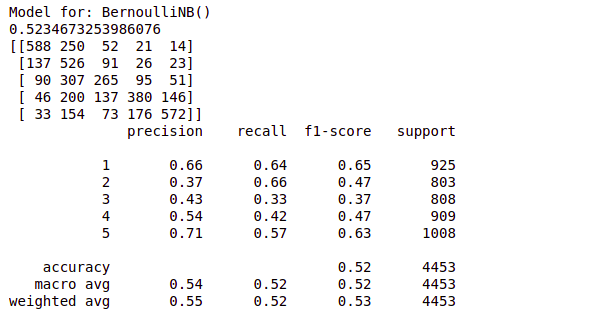
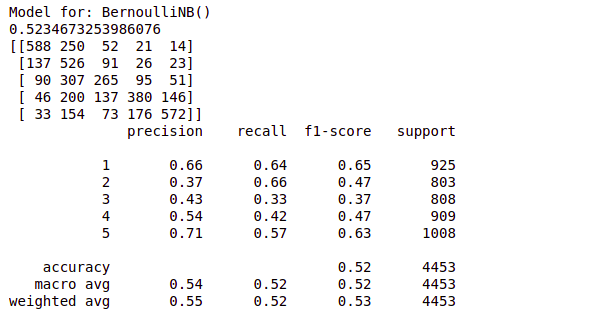
* KNeighborsClassifier()
* LogisticRegression()
* BernoulliNB()
* DecisionTreeClassifier()
* RandomForestClassifier()

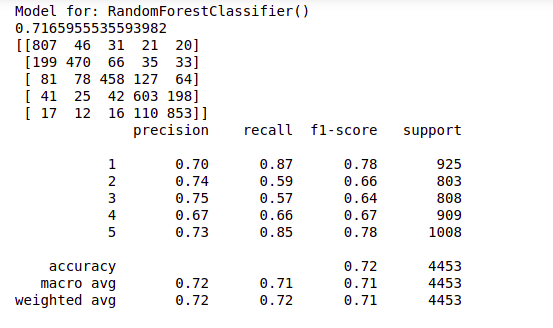
I applied all these algorithms in the dataset.

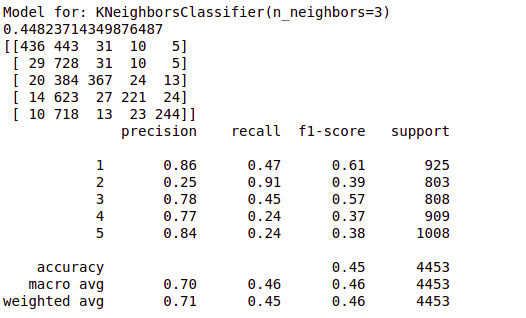
## Run and Evaluate selected models

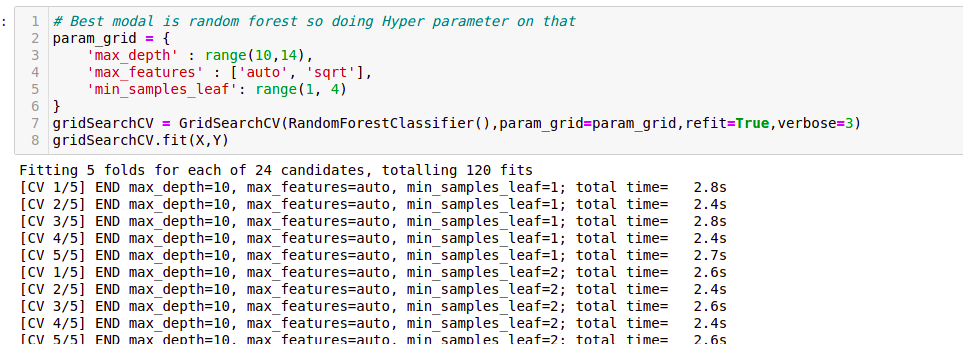








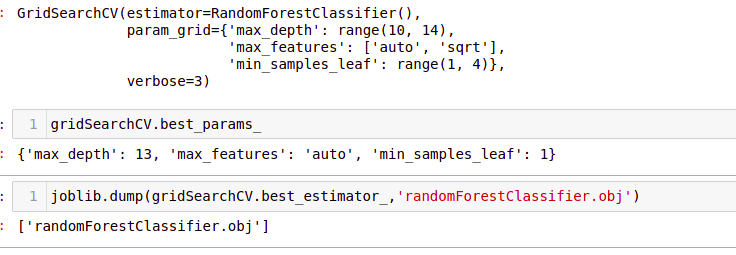




## Key Metrics for success in solving problem under consideration

Precision: can be seen as a measure of quality, **higher** **precision** means that an algorithm returns more relevant results than irrelevant ones.

**Recall** is used as a measure of quantity and high recall means that an algorithm returns most of the relevant results.



**Accuracy score** is used when the True Positives and True negatives are more important. **Accuracy** can be used when the class distribution is similar.

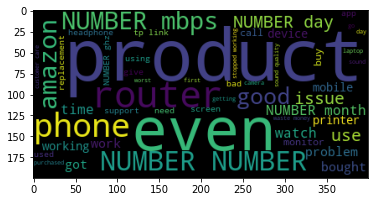
## Visualizations

Word Clouds -

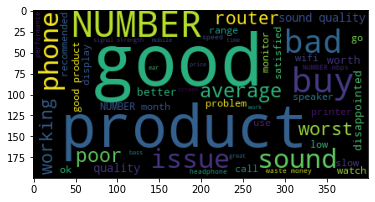
One Ratting( Review Title)



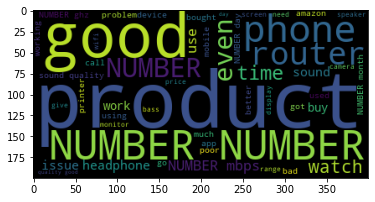
One Ratting( Review Desc)



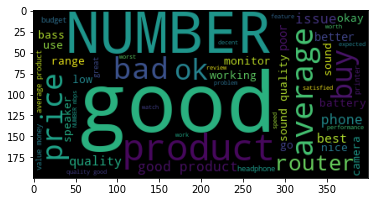
For 2 rattings ( Review Title)



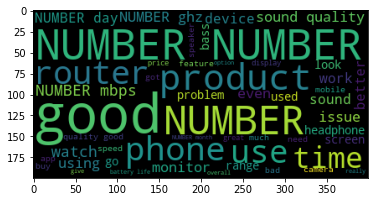
For 2 Ratting ( Review Desc)



#### For 3 rattings( For Title)



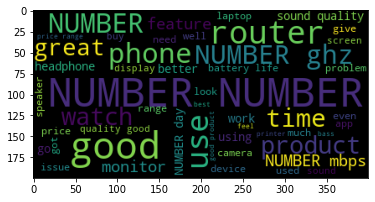
For 3 Ratting ( For Desc)



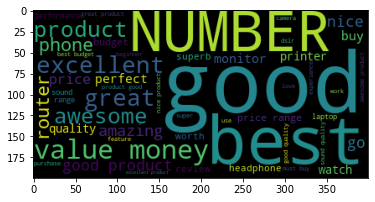
#### For 4 rattings[( For title)](http://localhost:8888/notebooks/11_ratings_project/ratting_prediction.ipynb" \l "For-4-rattings)



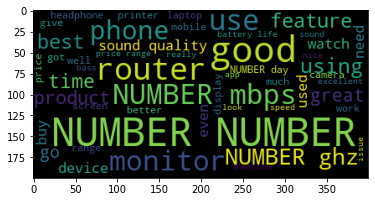
For 4 Ratting ( For Desc)



#### For 5 rattings(For title)



#### For 5 rattings(For Review)



Threat

Abuse

## Interpretation of the Results

From Random Forest Classifier R2 score was 71.64%,

Higher the R2 score means the model is well fit for the data. However, if R2 score is very high, it might be a case of overfitting. Other metrics Mean Absolute Error, Mean Squared Error and Root Mean Squared Error, with gradient boosting these scores are less then compared to other models. If these errors are less that means the model shows less errors.

# Conclusion

## Key Findings and Conclusions of the Study

First, we scrap Reviews and Ratings data of different technical products from flipkart.com using selenium web driver. After scrapping, we save this dataset in a csv file named “flipkartreviews.csv”. Then we pre-process data by cleaning duplicates, noise and unnecessary words which are not helpful for this project. Then we convert text into vectors by using tfidfvectorizer as we know our ml model understands only integers. We choose our best random\_state. Then we build our model. Though our dataset is imbalanced, we clearly see that f1 score of SVC is good compared to all other algorithms. So, we check croos\_val\_score of this model to check if it is overfitting or not. We confirm and conclude that this is the best model. We performed hyperparameter tuning by using GridSearchCV. We save the model by using joblib

## Limitations of this work and Scope for Future Work

As we know data is increasing in every second in our day today life. So more the data better the model. If we make this dataset for sentiment analysis, we choose ratings 3 or more as our threshold for being helpful reviews or good or positive reviews and below 3 we choose reviews is not helpful or bad reviews or negative reviews. For example: two people give their reviews on a product as ‘a nice product. But their ratings are ‘4’ and ‘5’ respectively. This is where the model fails to predict whether to choose rating ‘4’ or’5’ Due to increase in data in our daily basics, this model can be used to predict ratings of reviews. It might be a good tool for online shopping sites and manufacturer companies who may predict their customers ratings so that they can make their investment according to the demand of customers, which might help them to save time and earn more profits.