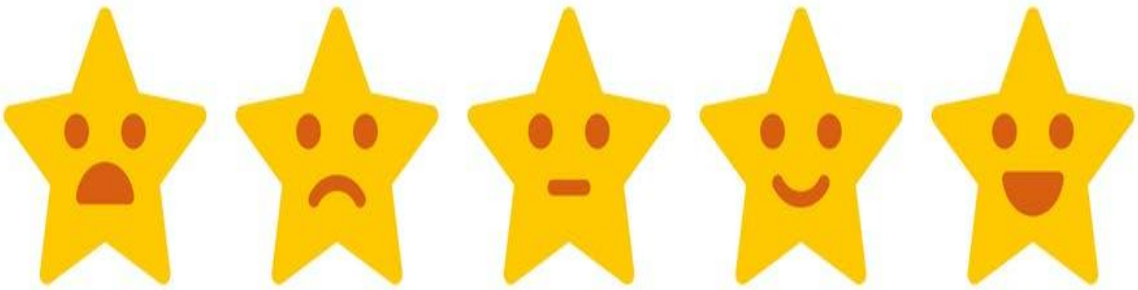
FLIP ROBO

RATINGS PREDICTION PROJECT



Submitted by:

Durgadhar Pathak

Internship 15

ACKNOWLEDGMENT

The internship opportunity I have with Flip Robo Technologies is a great chance for learning and professional development. I am also grateful to our SME Mr. Sajid Choudhary and Ms. Sapna Verma for their valuable and constructive suggestions during the planning and development of this project. Their quick support and references helped a lot in building this project. Also, I am very thankful to our SMEs & Flip Robo Team for understanding technical issue faced by me and provide quick resolution & providing enough time for submission.

Also, I am thankful to DT support Team for their continuous effort to resolve our queries during project building.

Research papers that helped me in this project was as follows: -

1- https://www.researchgate.net/publication/304757538_Review-Based_Rating_Prediction
2- https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00395-6
3- https://www.academia.edu/Documents/in/review_rating_prediction

Articles that helped me in this project was as follows:

1- https://link.springer.com/article/10.1007/s10462-020-09873-y
2- https://towardsdatascience.com/review-rating-prediction-a-combined-approach-538c617c495c
3- https://www.sciencedirect.com/science/article/abs/pii/S0306437921000132

References:

1- https://machinelearningmastery.com/
2- https://scikit-learn.org/stable/
3- https://www.geeksforgeeks.org/machine-learning/
4- https://pandas.pydata.org/
5- https://www.datacamp.com/
6- https://www.ibm.com/cloud/learn/machine-learning
7- https://www.selenium.dev/selenium/docs/api/py/common/selenium.common.exceptions.html
8- https://www.oreilly.com/library/view/web-scraping-with/9781491985564/
9- https://ieeexplore.ieee.org/document/5967324

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.

- ## Conceptual Background of the Domain Problem

This project contains two phase-

1- Data Collection Phase: -

We have to scrape at least 20000 rows of data. We can scrape more data as well, it's up to us. 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 can fetch other data as well, if we think data can be useful or can help in the project. It completely depends on our imagination or assumption.

2- Model Building Phase: -

After collecting the data, we need to build a machine learning model. Before model building do all data pre-processing steps involving NLP. Try different models with different hyper parameters and select the best model.

- Follow the complete life cycle of data science. Include all the steps like-

1. Data Cleaning
2. Exploratory Data Analysis
3. Data Pre-processing
4. Model Building
5. Model Evaluation
6. Selecting the best model

- # Review of Literature

First, we need to scrape the reviews of different laptops, Smart Phones, Headphones, smart watches, Professional Cameras, Monitors, Home theatres, Router from different e-commerce websites amazon, flip kart etc. using web scraping techniques and then need to build a machine learning model. Machine learning algorithms enable the creation of a new model using existing anonymized historical data that would be used to train the model to make better predictions not only for ratings, but also for other variables. With use of good model, companies could predict the ratings easily. To mitigate the subjective part of the decision-making process, different scoring models are introduced to evaluate certain parameters that could affect the reviews and ratings.

Models used: -

1- Random Forest: -Random forests select a subset of features in each of its decision trees thereby reducing the bias (because of high importance of single feature) of the model. The final output will be the mode of the outputs of all its decision trees which has better results than decision trees (which can possibly overfit). Hence, we chose to start our classification with random forests.

Other models used are: - Decision Tree, Gradient Boosting, Ada Boost Classifier, Multinomial NB, K Neighbors Classifier, Bagging Classifier, Extra Tree Classifier.

Hyper Parameter tuning: - We have implemented all the above models using Grid search cross-validation techniques to choose the best hyper-parameters.

Evaluation Matrix: - Confusion Matrix, Classification Report, Feature Importance, ROC_AUC _curve, & Scores (Accuracy, F1, Learning) etc.

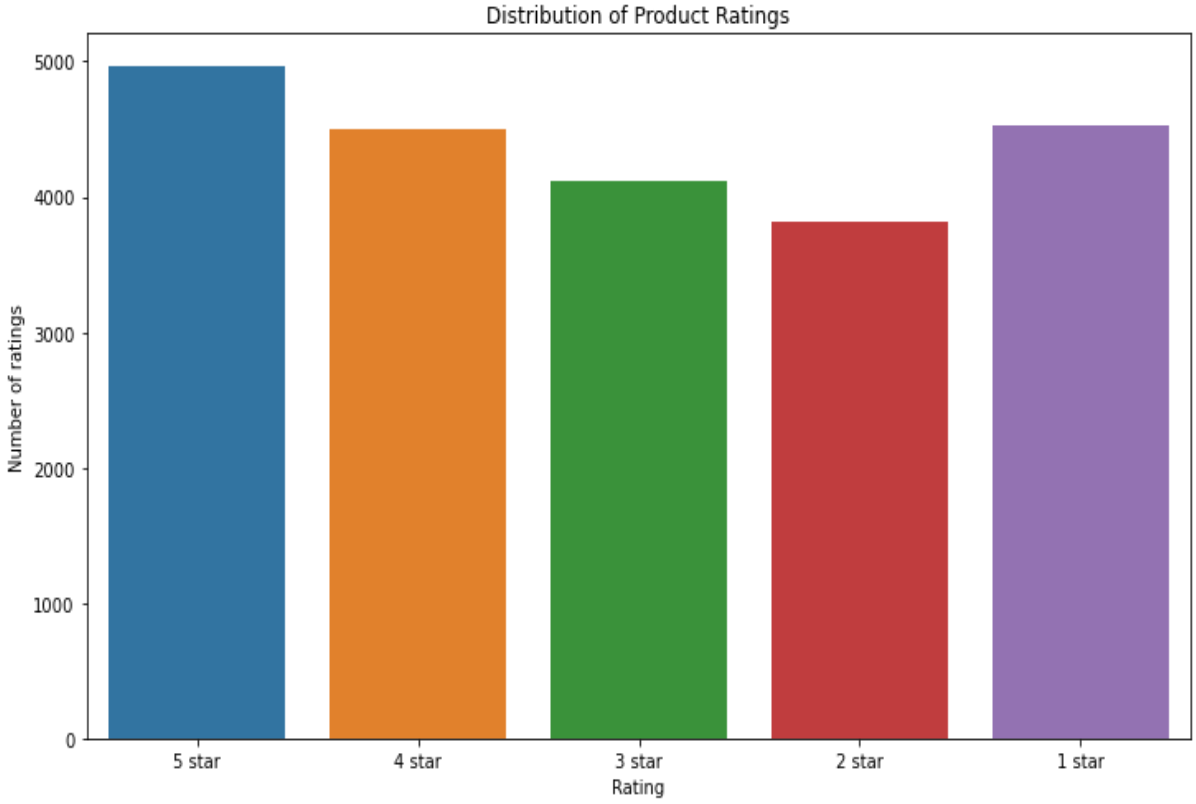
- # Motivation for the Problem Undertaken

Many product reviews are not accompanied by a scale rating system, consisting only of a textual evaluation. In this case, it becomes daunting and time-consuming to compare different products in order to eventually make a choice between them. Therefore, models able to predict the user rating from the text review are critically important. Getting an overall sense of a textual review could in turn improve consumer experience.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

In this project, input data is provided to the model along with the output data so it is a type of supervised learning. Also output Variable “Rating" is classification in nature so it is a Classification based problem and We have to predict the Ratings of the product with available reviews. We have performed classification tasks and it models a target prediction value based on independent variables and is mostly used for finding out the relationship between variables and forecasting. Data exploration is the first step in data analysis and typically involves summarizing the main characteristics of a data set, including its size, accuracy, initial patterns in the data and other attributes. There are 21926 rows and 4 columns. The distribution of product ratings shown as below: -

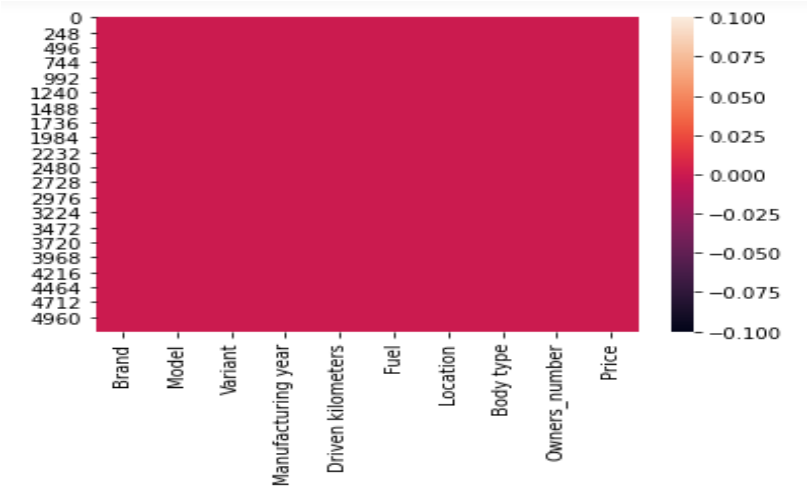


Rating numbers are as: -

```
#checking rating numbers detail
Rating.Rating.value_counts()
```

```
5 star    4960
1 star    4534
4 star    4496
3 star    4117
2 star    3819
Name: Rating, dtype: int64
```

Missing Data: -



Dataset has no missing values.

- # Data Sources and their formats

First, we need to collect review data from different websites using web scraping techniques and then need to build a machine learning model. So, we have scraped review data from different websites such as: - www.amazon.in using Selenium web scraping methods. We have scraped following data: -

Dataset: - Different columns are as: -

1) reviews of the product.
2) rating of the product.
3) Product Type
4) Product Title

We have combined data from this website into a csv file and used it for ratings prediction project. Total dataset has 21926 rows and 4 columns. We have also tried to equalize the dataset format for better understanding and use.

Target Variable: - Target Variable is Rating in this project and it is classification based in nature so we will use classification algorithms to make our model.

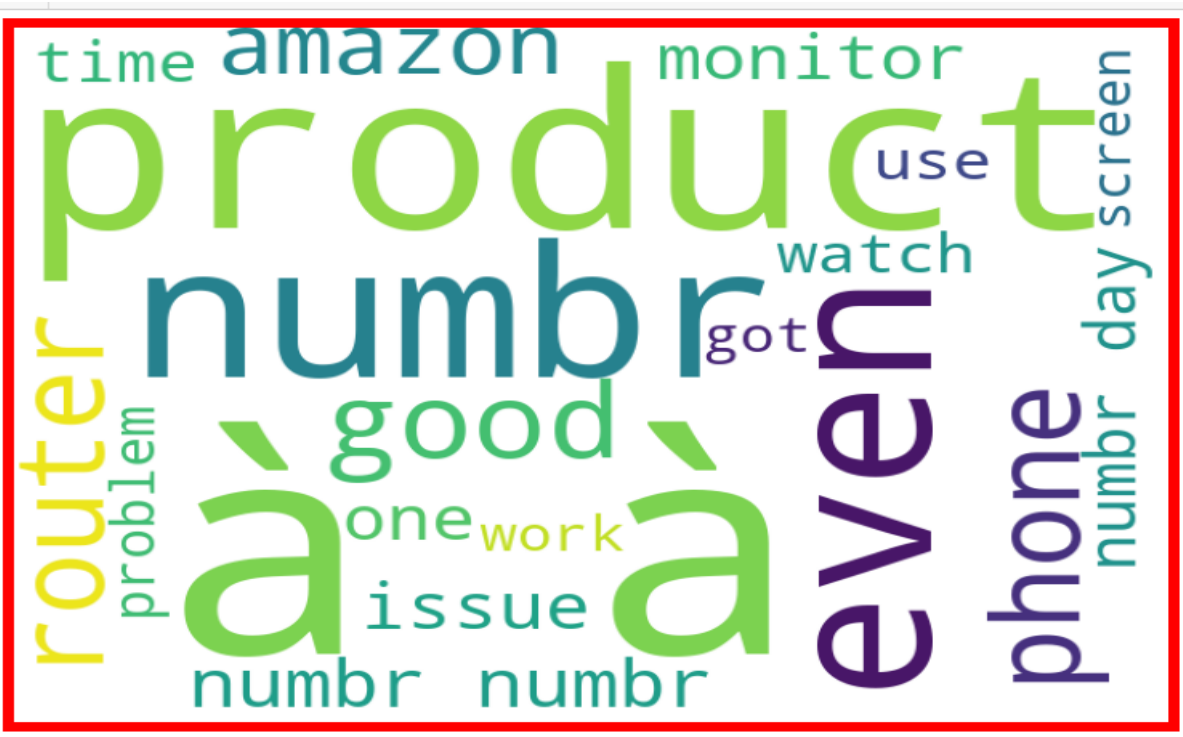
- # Data Pre-processing Done

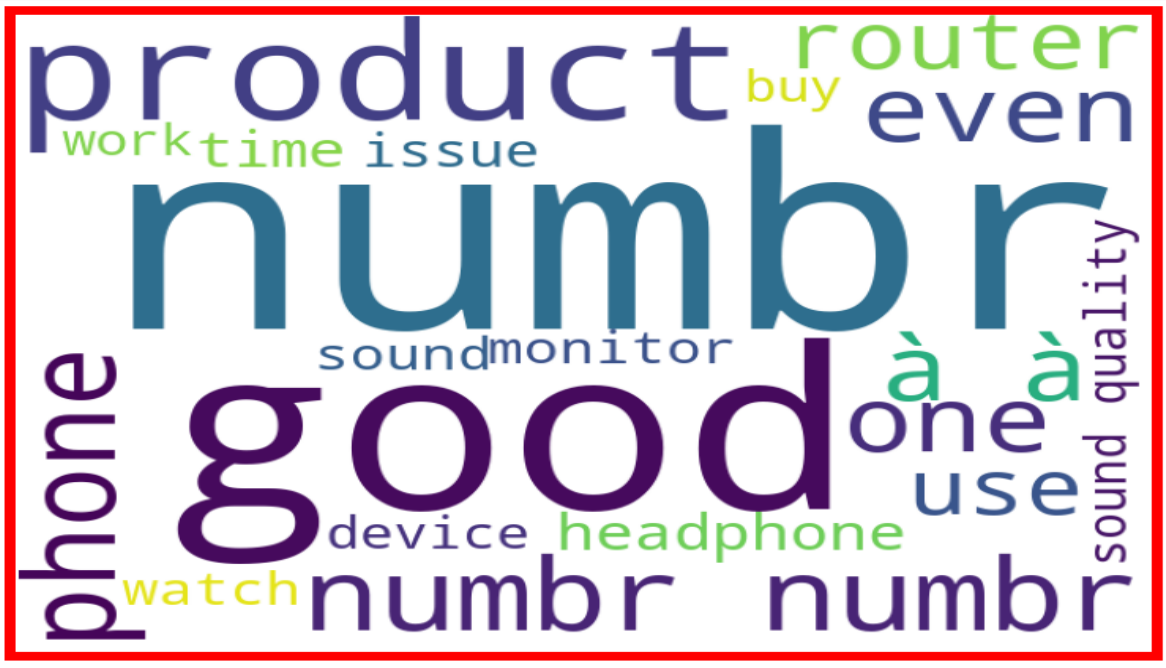
The data pipeline starts with collecting the data and ends with communicating the results & Data pre-processing is a data mining technique which is used to transform the raw data in a useful and efficient format. It involves 4 steps: - Cleaning, Formatting, Scaling, and Normalization. While building a machine learning model, if we haven't done any pre-processing like correcting outliers, handling different formats, normalization and scaling of data, or feature engineering, we might end up considering those 1% of results that are false. Some steps performed in this project are: -

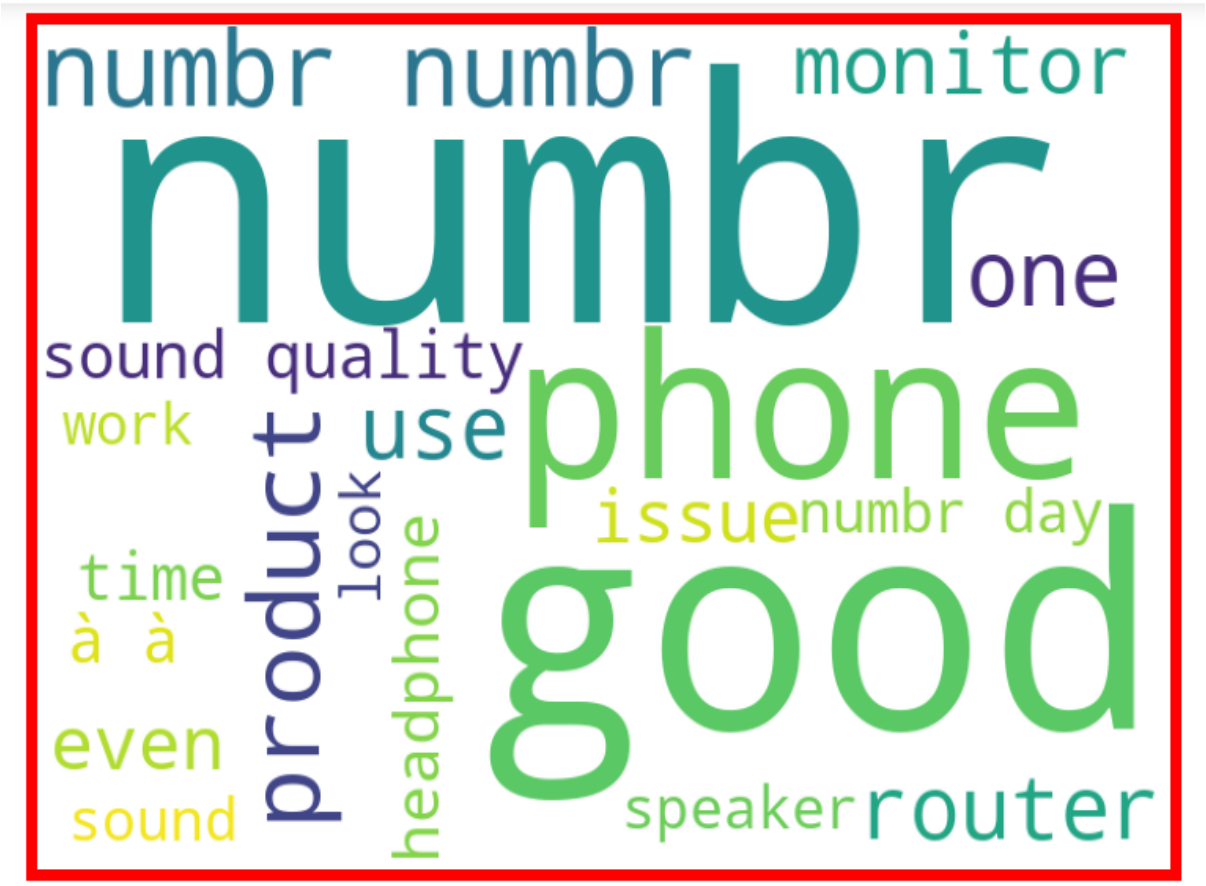
- Column wise Empty cell analysis done & found no missing values.
- Checked unique values in each column to explore dataset more deeply.

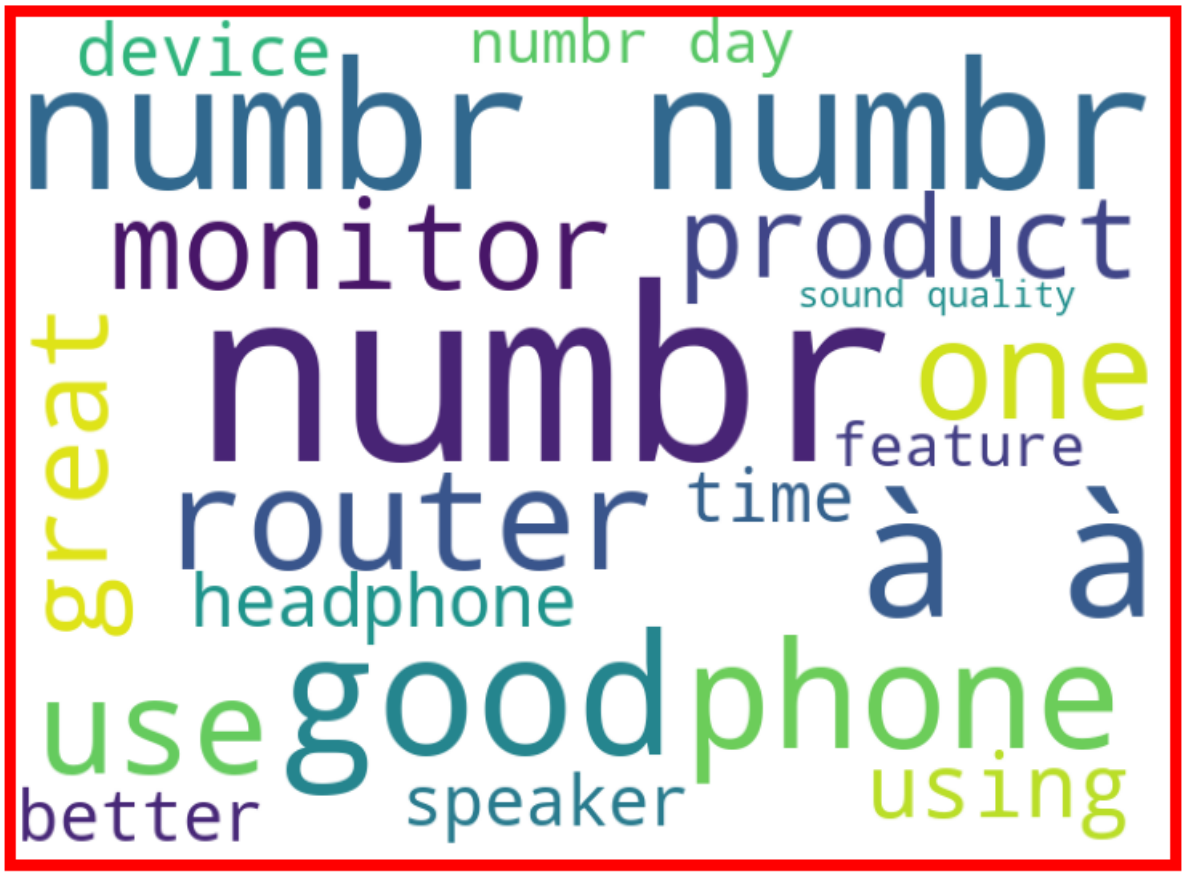
- **Feature Encoding and Normalization**: - Features came in a variety of format, e.g., integers, floating numbers, string values, etc. So, we Checked Concise Summary of our Data Frame and we have noticed that 8 columns have object (str or mix str) data type. This was a challenge for us as these features cannot be directly used for training. To prevent regression biases towards certain features, we dropped some of the irrelevant features and make sure that we haven't lose important data. In the end, we also normalize the feature values so that all features are evaluated on the same scale.

  Word Cloud for reviews (loud words) are as: -






numbr numbr
monitor
numbr
one
sound quality
phone
work
use
product
look
issue
numbr day
time
headphone
good
à à
even
sound
speaker
router


device
numbr day
numbr numbr
monitor
product
sound quality
numbr
one
great
feature
router
time
à à
headphone
good
phone
use
speaker
using
better

- Data Inputs- Logic- Output Relationships

Output Feature: - In this project Target Variable = dependent variable = y and shape of our dependent variable is (21926,1). The head of output feature is as: -

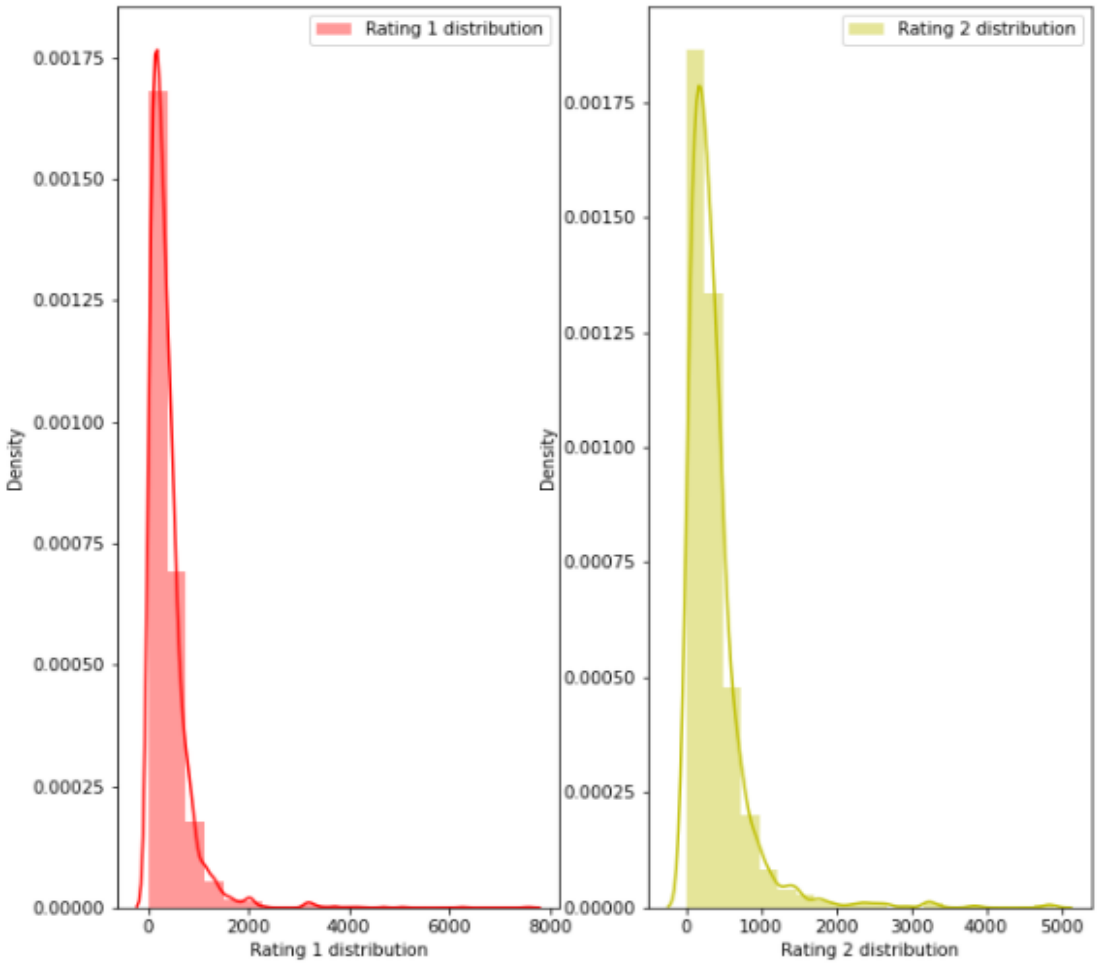
```
y=Rating['Rating']
```

Output Variable “Rating" is classification based in nature so it is a classification-based pro blem and We have to predict the Ratings with available independent variables.

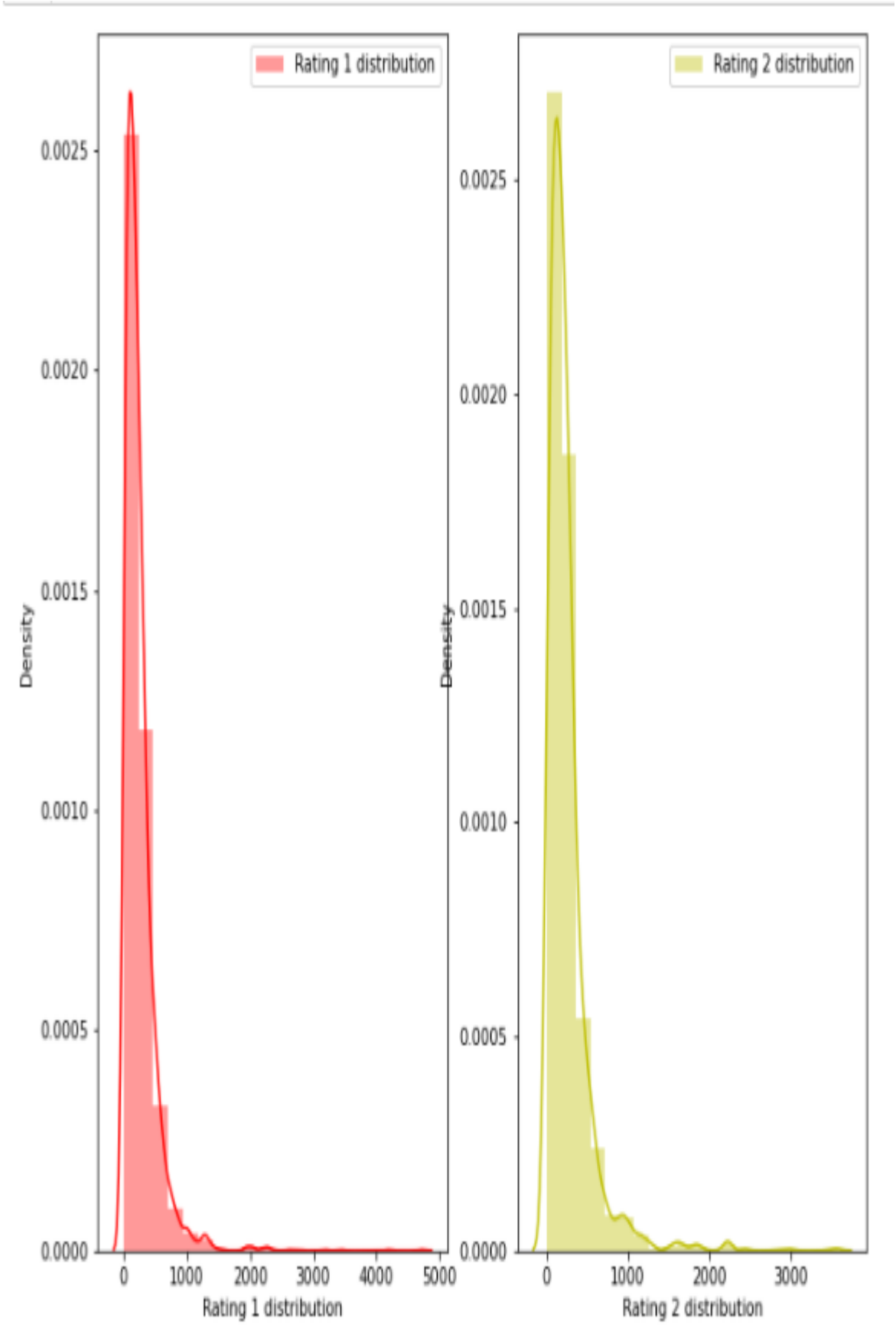
Input Feature: - The Independent variable in this project = feature vector = x and shape of Independent Variable is 21926 rows × 3 columns. We have merged Title features and Review feature in one column and also removed product type feature as it was not requir ed for our project.

Data Visualization: Data visualization is the graphical representation of information and data. Different visualizations for our input and output features are as: -

Distribution of Rating before data cleaning as: -



Distribution of Rating after data cleaning as: -



- State the set of assumptions (if any) related to the problem under consideration

- By looking into the target variable "Rating", we assume that this project is a classification-based problem as target variable is classification based in nature.
- We have removed irrelevant column which does not have more impact on dataset.

- Hardware and Software Requirements and Tools Used

Hardware:4GB RAM, Intel I3 Processor.
System Software: 64Bit O/S Windows 10(x64-based processor)

Software Tools: Software Tools used in this project are as: -

1- Anaconda3 (64-bit)
2- Jupyter Notebook 6.1.4
3- Python 3.8
4- MS-Office 2019 (Excel, Word, Power point)
5- Notepad
6- Google Chrome Web Browser
7- Selenium Driver

- Libraries & Packages used:

We have used Python and Jupyter Notebook to compute the majority of this project. For analysis, visualization, statistics, machine learning & evaluation, we have used these: -

1- Pandas (data analysis)
2- NumPy (matrix computation)
3- Matplotlib (Visualization)
4- Seaborn (visualization)
5- Scikit-Learn (Machine Learning)
6- SciPy (Z-score)
7- Selenium Web driver & Exceptions
8- Warnings (filter warnings) & etc. Microsoft Excel (for calculations and Data Handling).

Packages and libraries used in this project are: -

1- For Data Analysis & Visualization: -

```
#Importing Libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

2- For NLP: -

```
#importing NLP libraries
import re
import nltk
import string
from gensim.models import Word2Vec
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

3- From Scikit-Learn Library: -

```
#splitting the data into training and testing data

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.20,random_state=9)
```

```
#Importing all the model library

from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB

#Importing Boosting models
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import ExtraTreesClassifier

#Importing error metrics
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score,roc_curve,auc
from sklearn.model_selection import GridSearchCV,cross_val_score
```

4- For saving the final model: -

```
import joblib
```

- Testing of Identified Approaches (Algorithms)

1-RandomForest Classifier- Random forests select a subset of features in each of its decision trees thereby reducing the bias (because of high importance of single feature) of the model.

2-Decision Tree Classifier

3-KNeighborsClassifier(n_neighbors=6)

4-GradientBoosting Classifier

5-AdaBoost Classifier

6-Multinomial NB ()

7-Bagging Classifier ()

8-ExtraTreesClassifier ()

- **Key Metrics for success in solving problem under consideration: -**

	Model	Accuracy_score
0	KNeighborsClassifier	49.384405
1	DecisionTreeClassifier	68.080255
2	RandomForestClassifier	73.552212
3	AdaBoostClassifier	46.055632
4	MultinomialNB	57.843137
5	GradientBoostingClassifier	56.201550
6	BaggingClassifier	70.041040
7	ExtraTreesClassifier	72.777018

Metrics used: - 1-Confusion Matrix 2-Classification Report 3-Feature Importance

4-Roc_Auc_curve 5-Scores (Accuracy, F1, Learning) etc.

It's a classification problem and ROC_AUC curve metrices is used to observe True Positive Rate and False Positive Rate for users who have paid the loan and falsely they were marked as default and will also affect their credit score. We already talked about the importance of that in financial sector, and for the users who are marked falsely marked as paid but they didn't, can affect the company revenue.

Hyperparameter Tuning: - Hyperparameter tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. We used Grid Search CV to find estimators /neighbors/alpha for learning algorithms and Randomized search CV to find best parameters for implementation of final selected model.

Hyperparameter Tuning using Grid search trains the algorithm for all combinations by using the two set of hyperparameters (learning rate and number of layers) and measures the performance using “Cross Validation” technique. This validation technique gives assurance that our trained model got most of the patterns from the dataset.

- Visualizations (machine learning): -

All visualizations done before applying machine learning algorithms are shown above. We have done visualizations for exploring output variable, input features, relationship & correlation between input and output features, here we are visualizing different classification model’s performances, scores & Hyperparameter Tuning, Final model selection, relation between true & predicted values & test dataset.

```
#RandomForesetClassifier with best parameters found after hyperparameter tuning using gridsearch CV
rfc=RandomForestClassifier(max_depth=100, min_samples_leaf=3, min_samples_split=8, n_estimators=1000)
rfc.fit(x_train,y_train)
rfc.score(x_train,y_train)
predrfc=rfc.predict(x_test)
print(accuracy_score(y_test,predrfc))
print(confusion_matrix(y_test,predrfc))
print(classification_report(y_test,predrfc))
```

```
0.6716826265389877
[[839  33  34  14  29]
 [269 312  80  39  60]
 [139  31 456  88 119]
 [ 55   6  47 487 296]
 [ 27   3  14  57 852]]
              precision    recall  f1-score   support

           1       0.63      0.88      0.74       949
           2       0.81      0.41      0.54       760
           3       0.72      0.55      0.62       833
           4       0.71      0.55      0.62       891
           5       0.63      0.89      0.74       953

    accuracy                           0.67      4386
   macro avg       0.70      0.66      0.65      4386
weighted avg       0.70      0.67      0.66      4386
```

After Hyperparameter Tuning->

	Model	Accuracy_score
0	KNeighborsClassifier	49.384405
1	DecisionTreeClassifier	68.080255
2	RandomForestClassifier	73.552212
3	AdaBoostClassifier	46.055632
4	MultinomialNB	57.843137
5	GradientBoostingClassifier	56.201550
6	BaggingClassifier	70.041040
7	ExtraTreesClassifier	72.777018

Observations: After comparing above 8 models, these 3 models are good: -

1- Random Forest Classifier

2- Extra Trees Classifier

3- Bagging Classifier

Selection: Random Forest classifier is giving best results so we are implementing this model in our project.

Now we are going to do Hyperparameter Tuning for Random Forest regression models using Grid Search CV approach for best model selection so that we will get best results after model implementation.

Observation:

Random Forest Regression gives best performance before Hyper Parameter Tuning & also after Hyper Parameter Tuning as compared to other models so we are going to implement Random Forest Regression model in our project.

After implementation of Random Forest regression in our model we are going to check predicted values, true values and relationship between them using visualizations and also Checking Scores and errors after model fitting.

Saving Final model: - We have saved final model using job lib in *.pkl format.

Loading the saved model

```
model=joblib.load('Ratings_Prediction.pkl')
model
```

```
RandomForestClassifier(max_depth=100, min_samples_leaf=3, min_samples_split=8,
                       n_estimators=1000)
```

Evaluate Predictions: -

```
#Testing our model
import sys
nums= model.predict(x_test)
np.set_printoptions(threshold=sys.maxsize)
print(nums)
```

```
[5 2 1 2 1 4 1 1 3 5 4 5 3 5 5 3 1 2 5 1 2 3 1 3 4 5 3 3 1 5 1 5 5 2 1 2 5
 1 1 1 5 5 5 2 1 4 5 5 4 4 2 4 1 5 4 1 1 5 1 1 5 1 1 5 5 5 4 1 5 3 4 5 3 4
 1 4 5 4 1 5 5 1 5 5 1 5 5 3 1 5 5 3 1 5 2 5 5 3 4 3 1 2 1 5 5 2 4 2 1 5 5
 4 1 3 4 5 5 1 5 1 1 3 3 4 1 3 1 4 4 5 2 4 4 5 2 4 3 4 1 2 5 4 3 1 1 1 1 1
 5 5 5 3 3 3 3 1 1 5 3 2 1 5 4 1 4 5 1 4 3 2 4 1 4 5 4 5 1 5 4 4 4 1 1 5 1
 1 5 2 1 1 1 4 1 5 4 5 5 4 4 1 5 5 5 2 5 5 4 4 1 1 5 5 1 2 1 1 1 2 1 1 5 1
 3 3 3 3 3 4 5 5 5 5 2 2 3 1 2 5 5 3 4 2 5 3 4 1 1 1 1 1 3 5 3 5 1 3 5 3 1
 1 4 5 5 3 4 2 3 2 1 5 4 5 1 4 5 1 3 5 5 2 5 1 4 1 5 1 1 1 4 2 1 4 5 2 5 3
 4 4 4 3 1 4 5 1 5 3 3 5 1 1 5 4 5 5 1 4 4 1 1 2 2 1 2 3 1 5 1 5 2 1 3 3 1
 5 1 5 5 5 5 3 1 1 3 4 1 2 5 2 5 1 1 5 1 1 1 4 5 1 1 3 3 4 1 2 3 1 4 1 3 5
 5 4 1 5 5 4 4 5 1 5 1 5 1 5 5 5 5 1 1 4 1 4 4 5 1 5 4 3 2 1 2 2 5 1 5 1 5
 1 1 5 5 1 4 5 1 1 5 1 1 5 5 1 3 5 1 5 1 3 5 5 3 5 1 2 2 1 5 5 1 1 5 1 1 1
 4 1 3 1 5 4 4 5 5 4 2 2 3 5 5 1 1 5 5 5 1 5 3 5 1 2 5 4 2 5 1 1 1 1 5 3 2
 1 5 4 4 4 2 5 1 5 4 3 5 5 5 5 5 2 4 2 1 1 4 1 1 5 4 2 5 2 2 1 5 1 1 2 3 4
 2 5 5 5 4 5 3 3 5 2 4 5 4 3 1 5 1 4 5 5 1 1 1 5 1 5 5 1 1 5 1 5 5 5 3 5 1
 5 3 4 1 5 1 1 3 5 1 1 5 1 5 3 2 1 1 1 1 4 1 2 1 4 5 1 1 4 4 2 4 5 1 2 1 1
 4 5 5 5 3 5 1 5 2 5 5 5 3 5 4 1 1 2 5 2 3 5 1 3 5 4 5 1 2 3 3 2 4 3 3 1 4
 3 1 1 1 5 1 1 5 5 1 1 5 3 3 1 1 5 1 4 1 2 1 5 5 3 1 1 4 4 3 1 5 2 2 4 3 5
 2 1 4 1 4 5 1 2 1 3 5 5 4 1 5 5 3 3 5 4 3 5 1 5 5 3 4 4 5 2 5 1 1 1 5 4 1
```

CONCLUSION

- Key Findings and Conclusions of the Study

- In this project, we demonstrated the use of machine learning algorithms on a very challenging dataset to predict Ratings. To achieve the best performance, we showed that data pre-processing, a careful selection of techniques of balancing dataset, handling missing values, performed data cleaning, using Natural Language Processing, performed classification algorithms are all very important. Random Forest & Extra Trees classifiers work quite well on our dataset, and the use of Bagging Classifier is also effective. In the future, we want to continue exploring more sophisticated learning algorithms and dimension reduction techniques to further improve model performance on this important prediction task.

- Also, we have tested these machine learning algorithms on 2 different PCs of different technical configurations and have also compared their performance using evaluation & error metrics & and then chose the best performing model.

- **Learning Outcomes of the Study in respect of Data Science**

Learning outcomes are as: -

1- Data cleaning is quite tedious task and also very time taking and while working on this project we had encountered multiple issues but after research, study and guidance we neutralized these issues and also cleaned data properly.
2- Using Data visualization, we can easily identify missing values. Also, we can identify the relation between target & other features using it. In this project we have used Matplotlib & Seaborn library for data visualization.
3- Random Forest Regression have worked best in terms of accuracy score and other errors are also less. Even it gave best parameters during hyperparameter tuning and we have used these parameters in our final model.

- Limitations of this work and Scope for Future Work

Following limitations and scope of future work are as: -

1- Selection of best random state to calculate the maximum accuracy of model is very time taking especially in case of Random Forest and Extra Trees Algorithms.
2- Hyperparameter tuning using Grid Search CV is very time consuming specially in prediction of best parameters for AdaBoost, Lasso & k-nearest neighbors. So, we used randomized search cv to get best parameters for random forest regression.
3- Dataset have few missing values so it took time & steps to handle this issue.
4- Size of dataset is huge so sometimes it was difficult to handle this dataset but after completion of this project we got enough confidence to handle big datasets.
5- There are some irrelevant columns in this dataset so we have tried our best to check and remove them and also tried not to lose important data.
6- In future we will work on more algorithms to make more efficient model.

||Thank you||