FINAL REPORT

INDIAN PRODUCT REVIEW ANALYSIS AND RATING PREDICTION

PROBLEM STATEMENT

The campaign of **"Vocal for Local"** has seen tremendous support from the people of India, however for the campaign to sustain it is important that the Indian Brands are able to meet the quality of the products in demand by the consumers.

By using the **"Indian Product Reviews"** dataset of the Amazon we will try to gather insights if the consumers are happy and satisfied by the product provided to them.

We will be doing some EDA to get the insights from data like which brands and products are doing well while which are not.

Along with EDA and market analysis of the product, we will build a model that will help us determine/predict the rating of a product based on the review given by the user.

DATA WRANGLING

The raw dataset of the **Indian Product Reviews** consists of 5 columns and 2782 rows.

Following are the 5 columns present -

1. asin – It is a unique product ID
2. Name – Contains the name of the product.
3. Date – Contains the date when the review was given
4. Review – Contains the review for the product.
5. Rating – Contains the rating of the given product.

4 reviews were missing from the dataset; hence the number of rows were reduced to 2778.

The name column has the names written in unique fashion that allowed me to extract the name of the brand from the product name and save it to a new column consisting the Brand Name for each product.

The name of the brands was changed to lowercase to avoid any duplicity.

Finally, the review and rating data columns were divided into train/test split for modeling purposes.

EXPLORATORY DATA ANALYSIS

In this part we will get various insights that the data is providing us with.

Out of the total reviews we found there to be 24 unique brands and 122 unique products.

Having a look at the rating distribution for products we can have a broad idea of how the products are doing in the market.

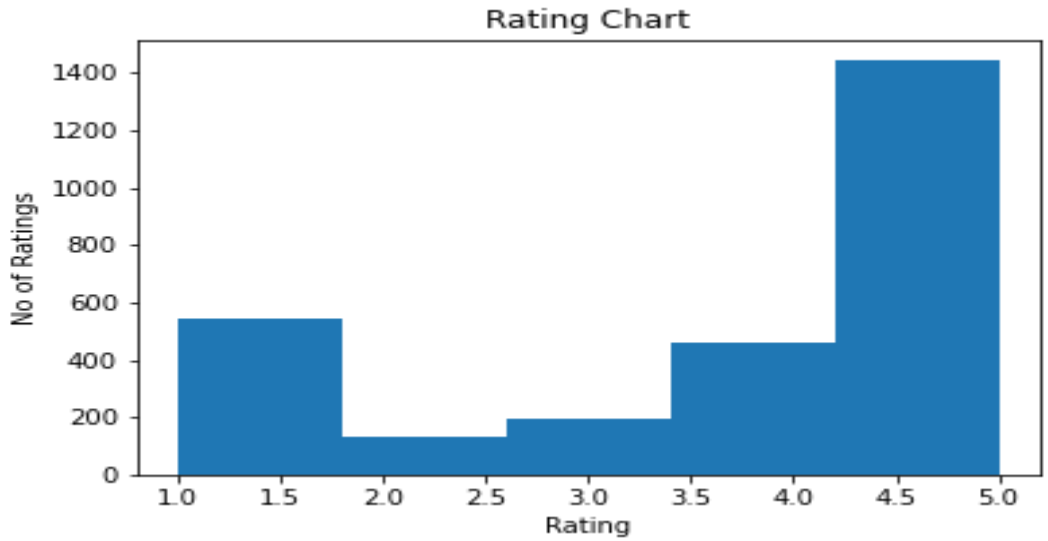


Fig 1: Rating Distribution

While the majority of products have a good rating, there is still a significant number of products having poor ratings.

Now, lets look at the top 5 brands that have the highest number of reviews.

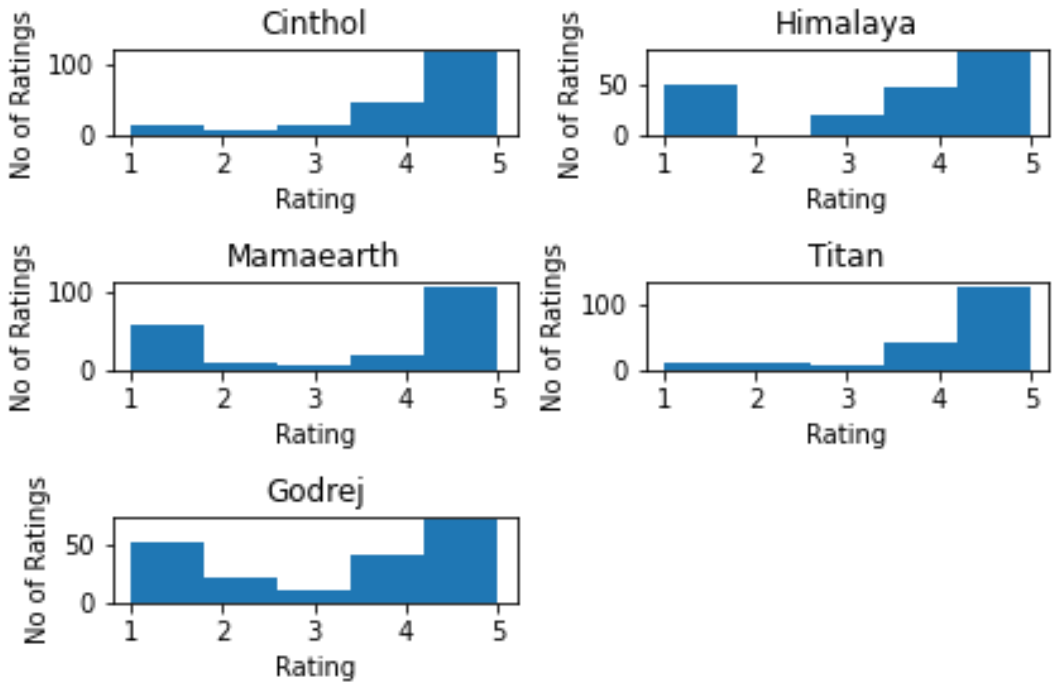


Fig 1.1: Ratings of Top5 Brands(No of reviews)

Bottom 5 Brands

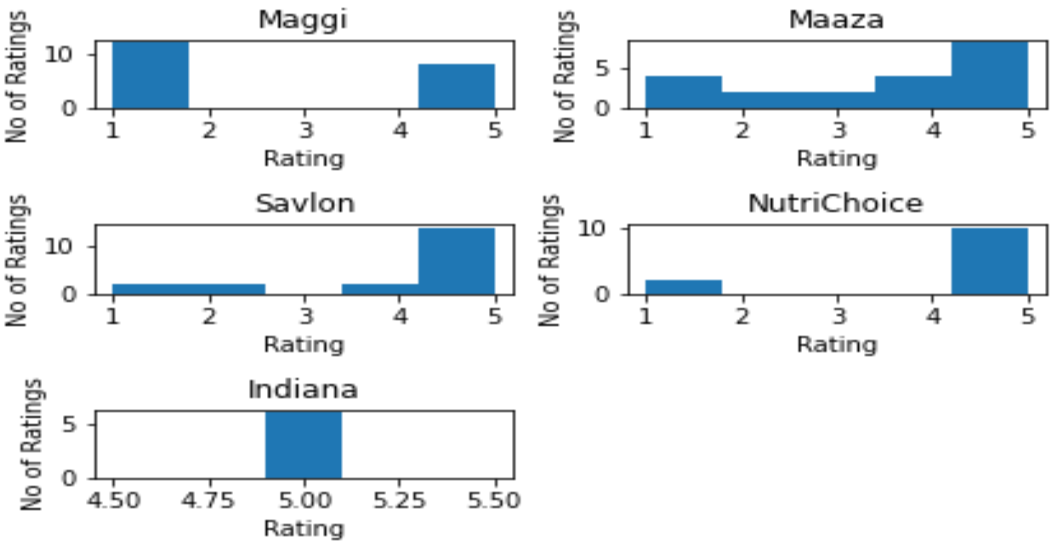


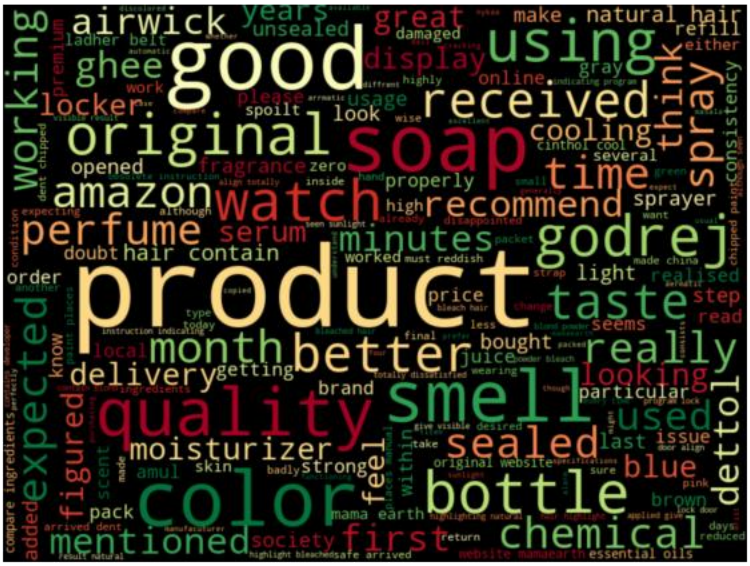
Fig 1.2: Ratings of Bottom 5 Brands (No of reviews)

WORDCLOUDS BY RATING

The following are the wordclouds for reviews of individual rating.

1 Star Review

2 Star Review



3 Star Review

Looking at the wordcloud we can iddentify that there are some common words that are present in every rating system.

The list of the common words that occur in every rating are-

```
['product', 'soap', 'amazon', 'smell', 'time', 'good', 'received',
'skin', 'taste', 'using', 'bought', 'quality', 'pack', 'fragrance',
'brand', 'color', 'look', 'watch', 'better', 'month', 'less', 'feel',
'though']
```

Pre-Processing and training

We will look at creating three different data sets for the purpose of training and see which is a better performing data set while modeling.

The three different data sets that we have prepared are-:

1) Clean data only removed of stop words.

LOCATION - /Data/Test-Train(Stopped Words)/train_clean.csv

2) Clean data removed of stop words and stemmed.

LOCATION - ../Data/Test-Train(Stemmed Words)/stem_train.csv

3) Clean data removed of stop words and the common words occurring in the reviews.

LOCATION - /Data/Test-Train(Common Words)/train_stop.csv

MODELLING

Before modelling we are going to transform our data using **Count Vectorizer and TF-IDF Vectorizer.**

We will then use the data set for each above-mentioned technique to train our model, the technique that gives better model results will be taken into consideration for further Modelling procersses.

MODEL1 (NAIVE BAYES WITH COUNT VECTORIZER)

DATASET USED - train_clean.csv (data with stop words removed)

TECHNIQUE – Count Vectorizer

Parameters - max_df=0.95

min_df=2

MODEL USED - Naïve Bayes

Parameters – Default

In this modelling technique we used count vectorizer on the stop word removed training data.

Once we have fitted the model with the training data set, we did prediction on the test set.

The Accuracy score we received was `0.7529976019184652` i.e the model was **75%** accurate.

Now we would change the dataset and see if the model performance has positive or a negative impact. Now we will be using stemmed data.

DATASET USED - stem_train.csv

After we fit the vectorized stemmed data in our model we used test data to predict.

The Accuracy score received for this model was **`0.7254196642685852`** `i.e the model was 72% accurate.`

`We saw a decline in the performance of model while using stemmed data for training.`

MODEL2 (TF-IDF)

DATASET USED - train_clean.csv (data with stop words removed)

TECHNIQUE – TfidfVectorizer

Parameters - max_df=0.95

min_df=2

MODEL USED - Naïve Bayes

Parameters – Default

In this model we used TF-IDF vectorization on our training set while keeping all the other parameters same as the prior model.

The Accuracy Score we received for this model was 0.6306954436450839 that is the model was **63% accurate.**

The performance is seen to be drastically decreasing by using the TF-IDF.

For the validation we used cross validation technique with 5 folds.

The mean of the scores was 0.6430008215620279, confirming the performance of our model.

We also used TF-IDF to check with stemmed data as well.

The accuracy score we got while training the model with the stemmed data 0.6354916067146283 that is the model was only accurate 63% of time.

Now that we have seen that our models are working better with count vectorized training set rather than a TF-IDF vectorized set. Therefor for the coming models we are going to use Count vectorized data.

MODEL 3(DECISION TREES)

DATASET USED - train_clean.csv (data with stop words removed)

TECHNIQUE – CountVectorizer

Parameters - max_df=0.95

min_df=2

MODEL USED – DecisionTreeClassifier()

Parameters - criterion="gini", random_state = 12

Using the Gini Impurity model of the decision trees we trained our model with the train data with removed stopped words.

The accuracy score we got using the Decision trees was far better than the Naïve bayes that is it got 0.8860911270983214 which is **88%** of accuracy.

Using the entropy model of the DecisionTreeClassifier we got the similar result.

Now we are going to use RandomForestClassifier to see if we could better the performance of our model..

MODEL 4

DATASET USED - train_clean.csv (data with stop words removed)

TECHNIQUE – CountVectorizer

Parameters - max_df=0.95

min_df=2

MODEL USED – RandomForestClassifier()

Parameters - n_estimators = 90,

n_jobs = -1,

oob_score = True,

bootstrap = True,

random_state = 42

Using random forest did not improve the model performance by a lot and the accuracy score remained around 89%.

For this reason, we are going to let Random Forest be our best model to be selected.

Now that we have finalised our model let us select look at the most important features and disregard those that have little or no effect.

The total number of features that were initially present in the data were 3424.

We looked at the feature importances and selected best features at the threshold of 0.001.

After feature selection the number of features in the training data was 216 while the performance of the model remain unaffected.

It was noticed that increasing the threshold to select lesser features impacted the model negatively.

CONCLUSION

In the given project we were given a set of Indian Product Reviews on amazon along with the ratings it received.

Our aim was to predict the ratings based on given review.

We cleaned and prepared our data into two different sets (Stopped and Stemmed) to see which performed better with the model. The stopped performed better and was chosen for further processes.

There was total three model techniques used - Naïve Bayes, Decision Trees and Random Forest.

Random forest came out to be the best performing model with 89% accuracy.