Task 1

To kick things off you can start by scraping Mamaearth's Beauty category: https://mamaearth.in/product-category/beauty Ideally you would want to scarpe the following basic data points:

- Product Name
- Product Link
- Rating
- Reviews
- MRP
- Pack Size
- Discount/Offers running on the product

We would like you to write a classification algorithm which can fetch the key ingredient used in the product and also classify the product based on its category. So, taking Mamaearth Onion Hair Oil as an example, the key ingredient here is Onion and the category is Hair Oil.

Approach and Logic

Following points discuss the approach used to solve the above problem statement:

- For the web scraping part, **Selenium Web Driver** is used. I have created functions to scrape the data points mentioned in the above problem statement.

Functions	Description
scrape_title()	Scrapes the product titles from the webpage.
scrape_product_link()	Scrapes the link for each product on the webpage
scrape_ratings_reviews_mrp_discount()	Scrapes ratings, reviews, mrp and discount of each product on the webpage.
scrape_descriptions()	Scrapes the description of the product from the webpage.
scrape_key_ingredients()	Scrapes the key ingredients in the product from the webpage.

- Next, I clean and transform the scraped data and store it in a **pandas dataframe**.

- After this I perform Exploratory Data Analysis on various features. Some of the insights are attached below.

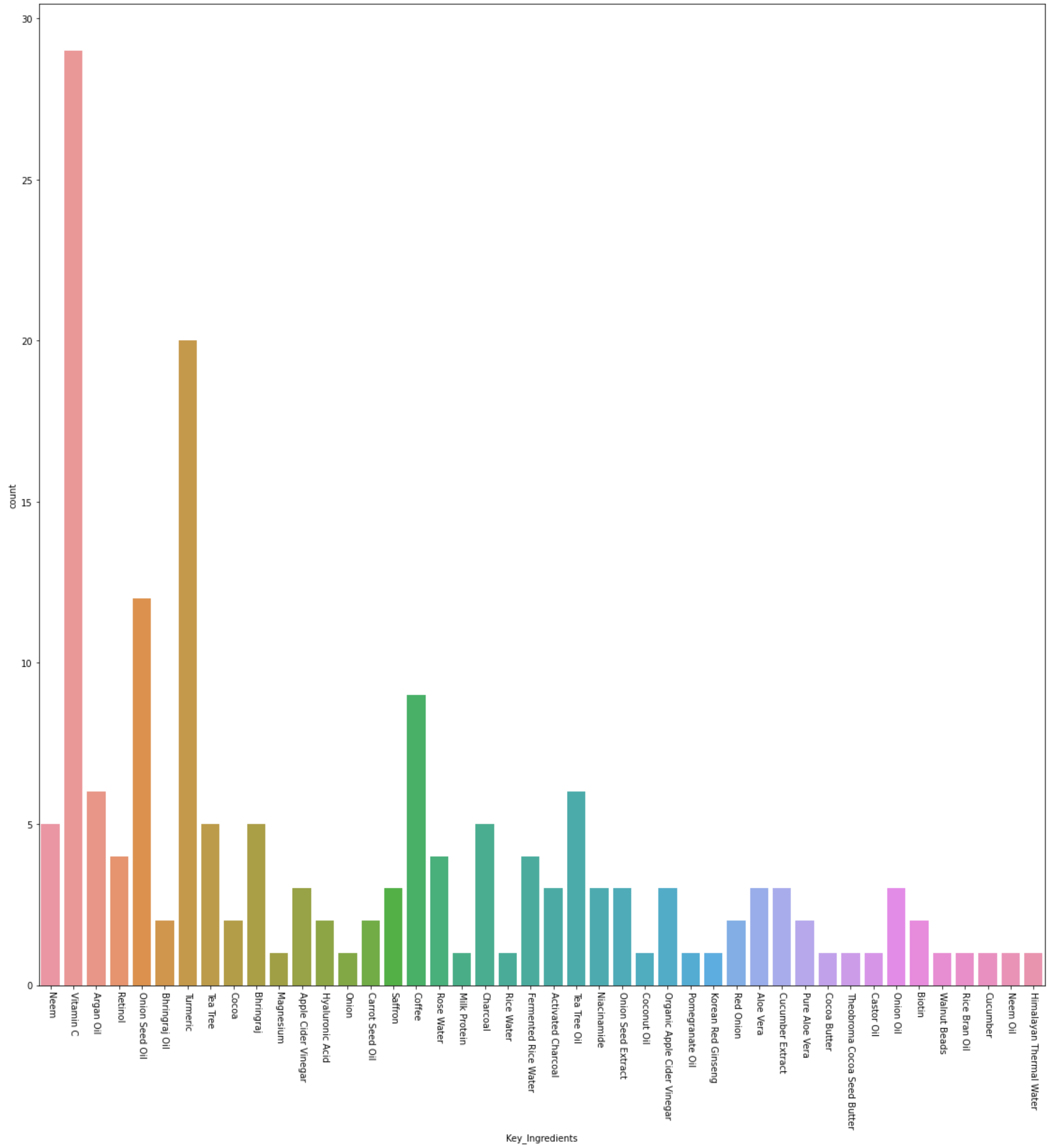


Fig 1. Count of each key ingredient

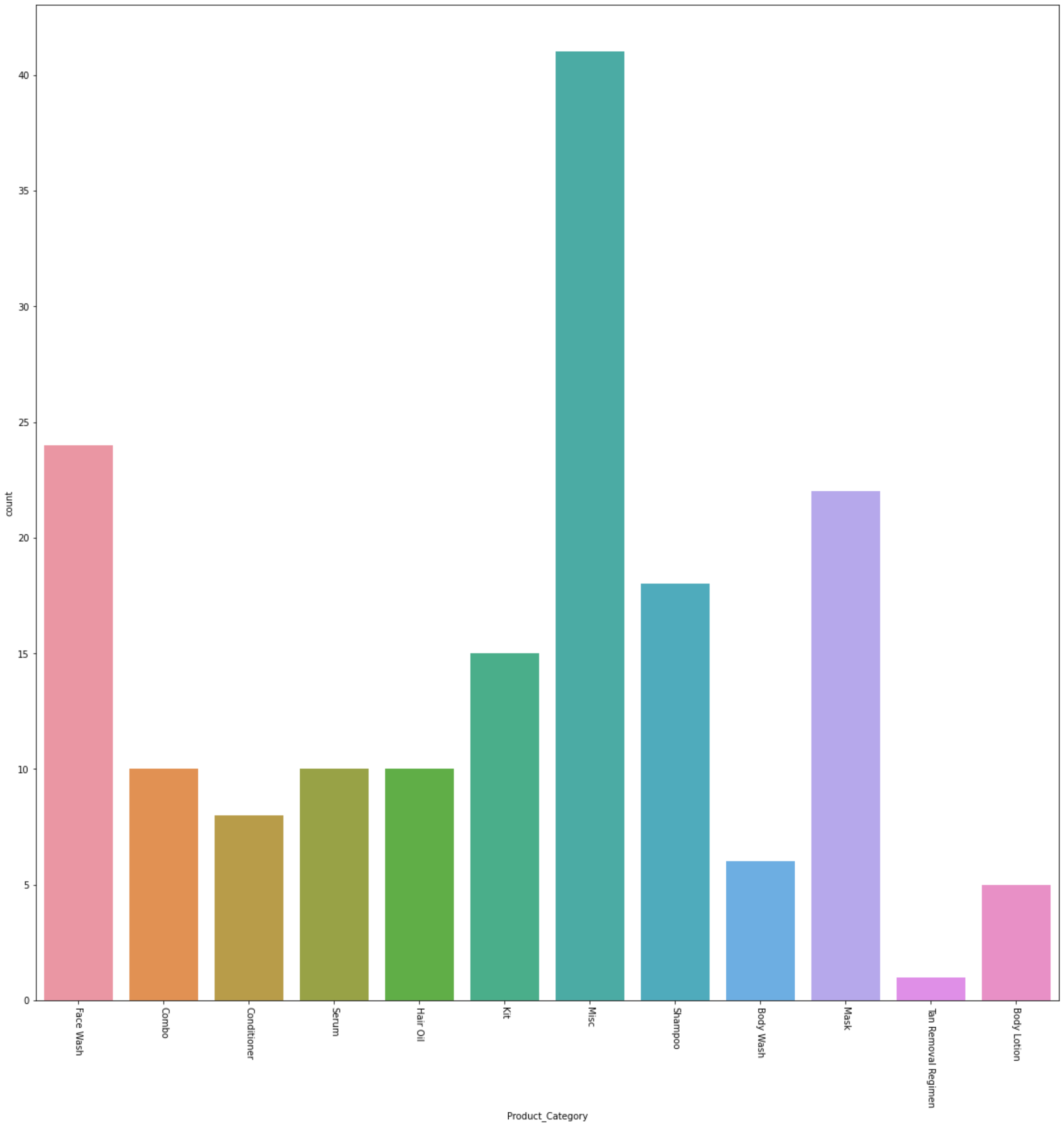


Fig 2. Count of each Product Category

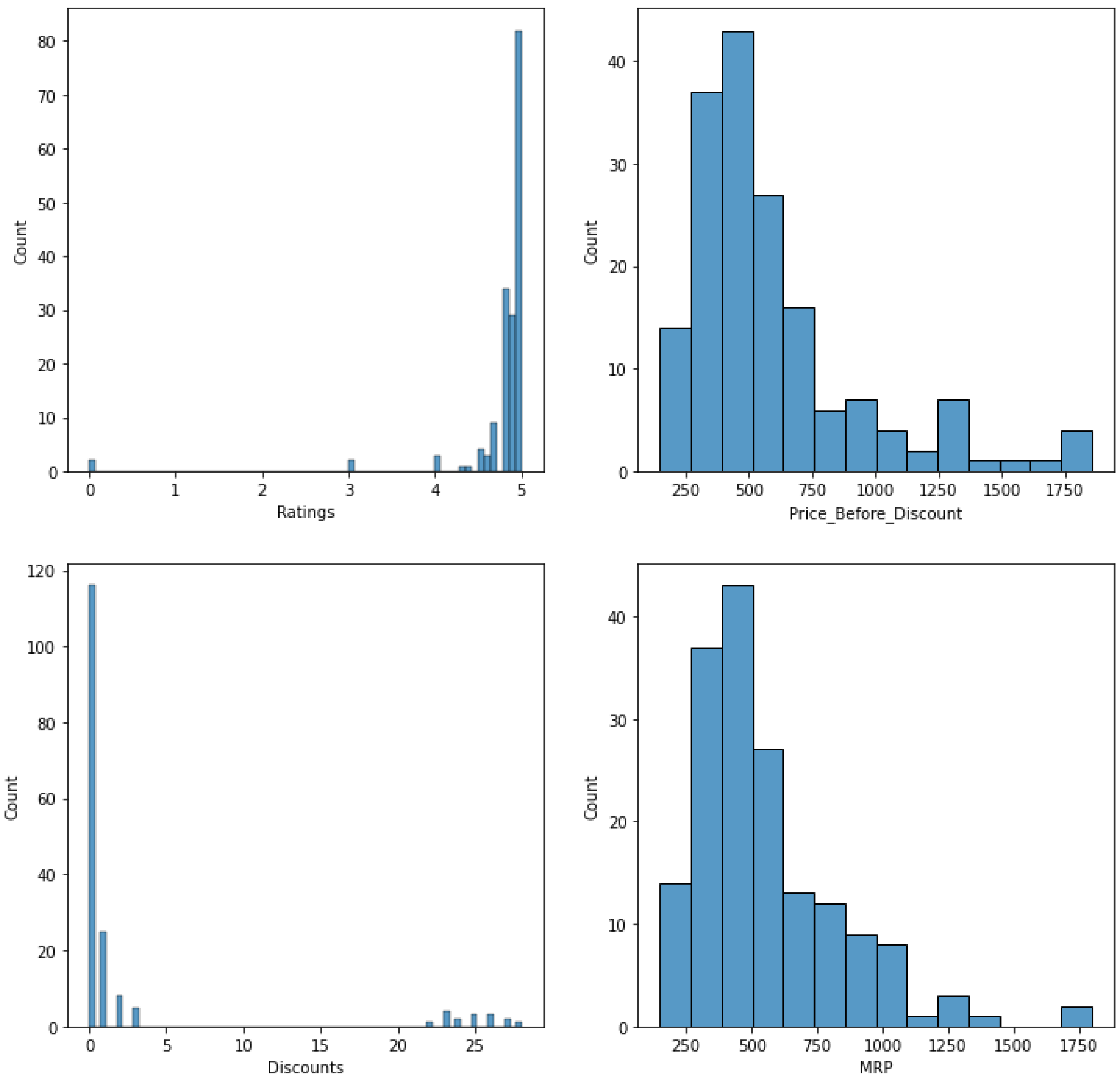
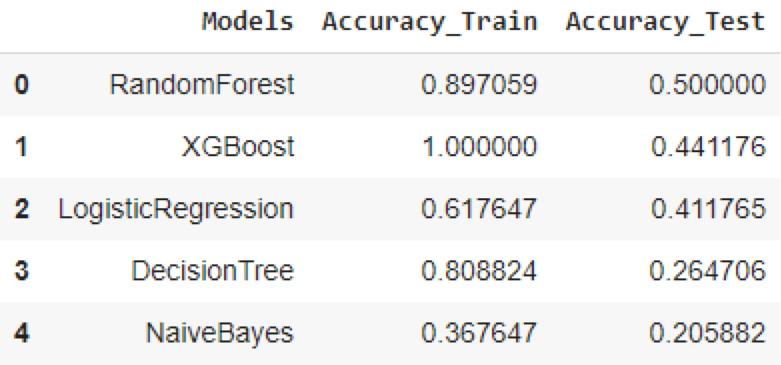


Fig 3. Frequency distribution of numerical features (Ratings, Price_Before_Discount, Discounts(%), MRP

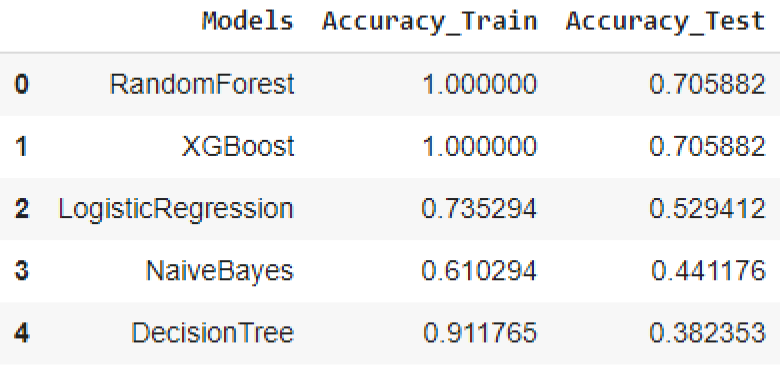
For all the insights, please refer to the attached Google Colaboratory Notebook.

- Next, for classifying Key Ingredients and Product Categories, data is prepared using **Bag Of Words Pipeline**.
- Various classification models are built to classify and Key Ingredients and Product Categories and their **accuracy** is compared. The results are stored in the attached excel sheet and also presented below.



	Models	Accuracy_Train	Accuracy_Test
0	RandomForest	0.897059	0.500000
1	XGBoost	1.000000	0.441176
2	LogisticRegression	0.617647	0.411765
3	DecisionTree	0.808824	0.264706
4	NaiveBayes	0.367647	0.205882

Performance Metrics (accuracy) of various classification models on train and test data for classifying Key Ingredients in the products



	Models	Accuracy_Train	Accuracy_Test
0	RandomForest	1.000000	0.705882
1	XGBoost	1.000000	0.705882
2	LogisticRegression	0.735294	0.529412
3	NaiveBayes	0.610294	0.441176
4	DecisionTree	0.911765	0.382353

Performance Metrics (accuracy) of various classification models on train and test data for classifying Product Categories.

- Next, the **collected data**, **predictions** and **performance metrics** are stored in an excel workbook.

Conclusion

Overfitting can be observed in both the classification tasks which can be due to the lack of data. **Increase in data** and use of **regularization techniques** will lead to significant increase in performance metrics.