INNOMATICS®
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

SENTIMENT ANALYSIS

About us

- Vijay sada
- B-tech in Electronics and Communications Engineering.
- Fresher
- LinkedIn - https://www.linkedin.com/in/vijay-sada/
- GitHub - https://github.com/vijaysada

- Dawa Phuti Lepcha
- MSc. Mathematics, Applied mathematics and computer programming
- Fresher
- LinkedIn - https://www.linkedin.com/in/dawa-phuti-lepcha-646768240/

Introduction

- Sentiment Analysis of Amazon reviews is aimed at deciphering the sentiments expressed by customers in their textual feedback and predicting corresponding scores. As one of the largest online marketplaces, Amazon accumulates a vast repository of customer reviews spanning a wide array of products.

- Understanding the sentiments conveyed in these reviews is essential for Amazon and its sellers to gauge customer satisfaction levels, identify product strengths and weaknesses, and make data-driven decisions to enhance overall customer experience.

- By leveraging these insights, businesses can drive continuous improvement, enhance product quality, and foster stronger customer relationships, ultimately bolstering their competitiveness in the dynamic landscape of e-commerce.

Data Overview

The dataset was provided to us by the client.

The dataset includes the following features:

- **idlist**: Unique row number.
- **productid**: The id of the product which is being reviewed.
- **Userid**: The id of the customer who reviewed the product.
- **ProfileName**: The name of the customer in the profile.
- **HelpfulnessNumerator**: The number of customers who found that review helpful.
- **HelpfulnessDenominator**: The number of customers who indicated whether they found the review helpful or not.
- **Helpfulness**: HelpfulnessNumerator/ HelpfulnessDenominator
- **Score**: The rating of the product from 1 to 5.
- **Time**: Timestamp for the review.
- **ReviewSummary**: Summary of the Reviews of the product given by the customers.
- **ReviewText**: The review of the product given by the customers.

The target column is ‘**Score**’

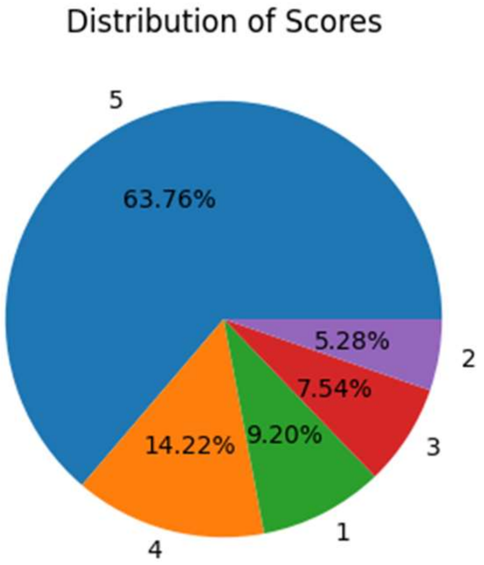
S No	Feature Name	Data Type
1	idlist	Int64
2	productid	Object
3	Userid	Object
4	ProfileName	Object
5	HelpfulnessNumerator	Int64
6	HelpfulnessDenominator	Int64
7	Helpfulness	Float64
8	Score	Int64
9	Time	Int64
10	ReviewSummary	Object
11	ReviewText	Object

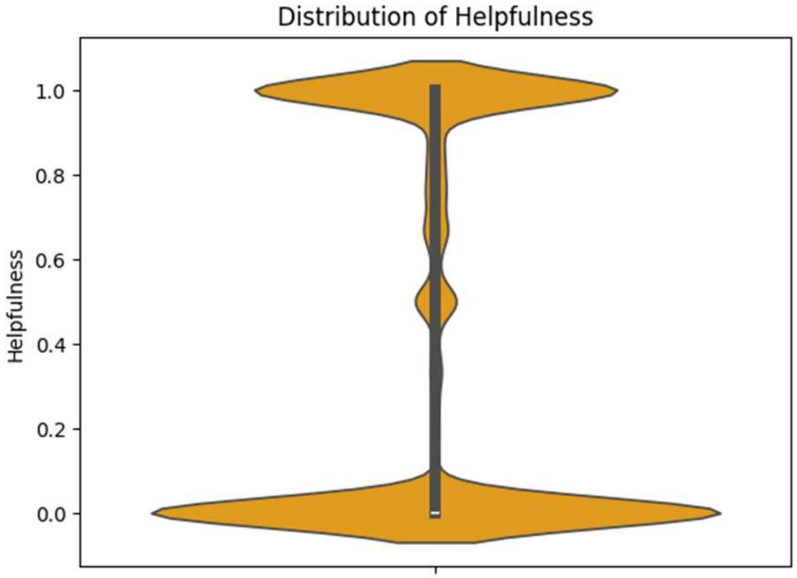
Pre-Processing

For the project, the following preprocessing steps were performed:

- **Data Cleaning:** Identified and handled inconsistencies like missing values, errors like broken html tags, and found 2 outliers were found in the dataset.
- **Lowercasing:** Converted all the text data into lower case so that there aren't any inconsistency in the data.
- **Handling Missing Values:** Dropped the missing values since they only contributed to 0.0249% of our total data.
- **Handling the duplicates:** Dropped the duplicates based on the reviewtext column and reviewsummary column separately.
- **Unmeaningful words and Stopwords:** Removed the stopwords and the broken html tags from the reviewtext column and reviewsummary column since they don't hold any meaning.
- **Lemmatization:** Used Lemmatization for pre-processing on texts to retain the true meaning of text.
- **TF-IDF & BoW:** Used both TF-IDF and BoW for representing the textual data as numerical vectors to ensure they ready to be use for the model training.

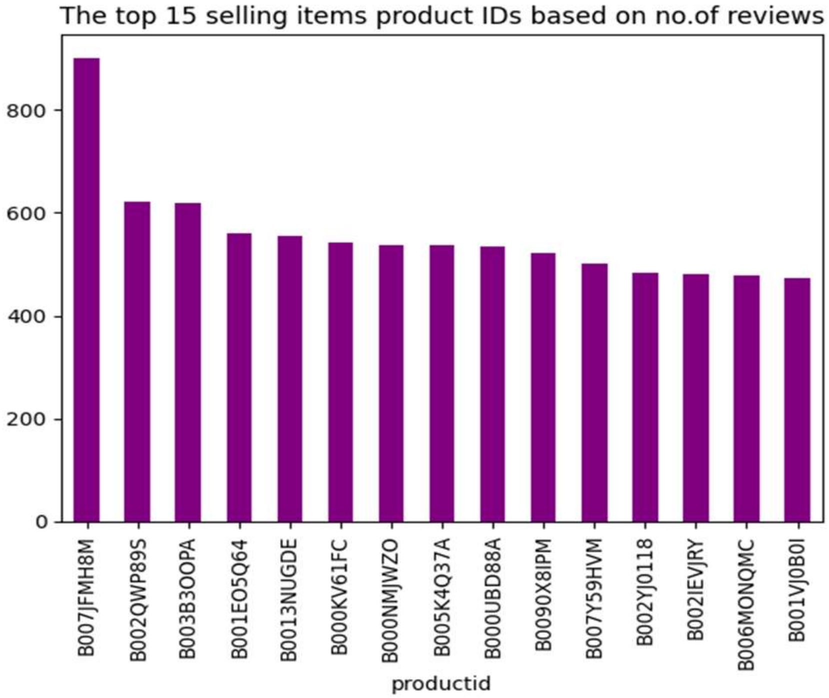
Exploratory Data Analysis (EDA)



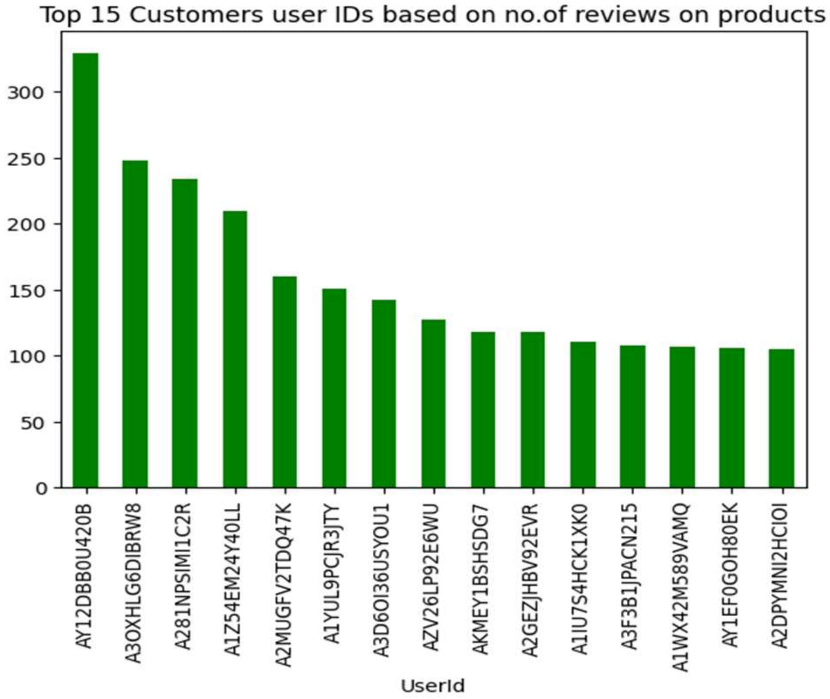


- Around 64% of the data has 5 Score i.e. There are far more number of good reviews compared to poor reviews. We can observe that the customers have given very less 2 score whereas majority of them have given the 5 score.
- From the above we can observe that either most of the reviews are greatly useful and some are not.

EDA (Cont.):

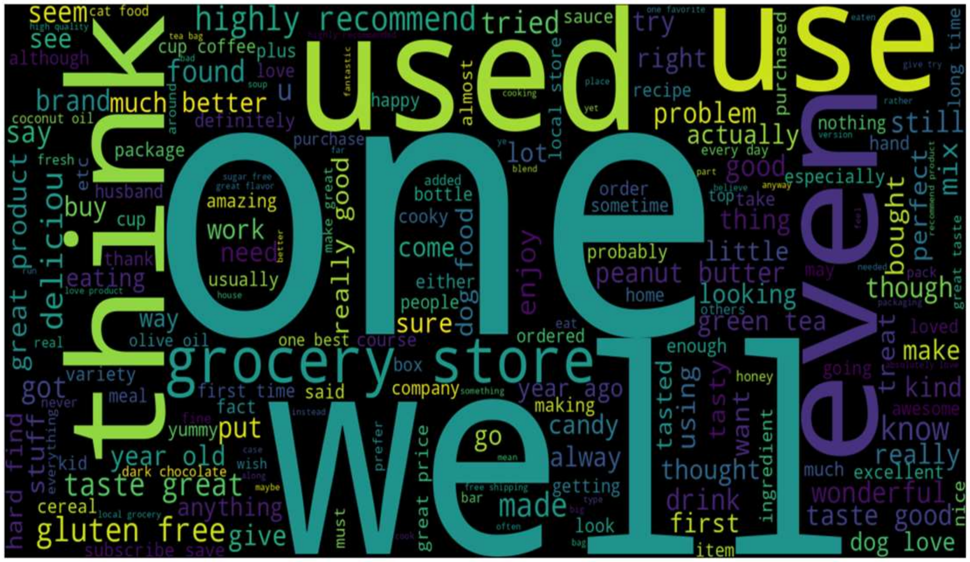


From the above we can observe the most selling items based on the no.of reviews left by the user.

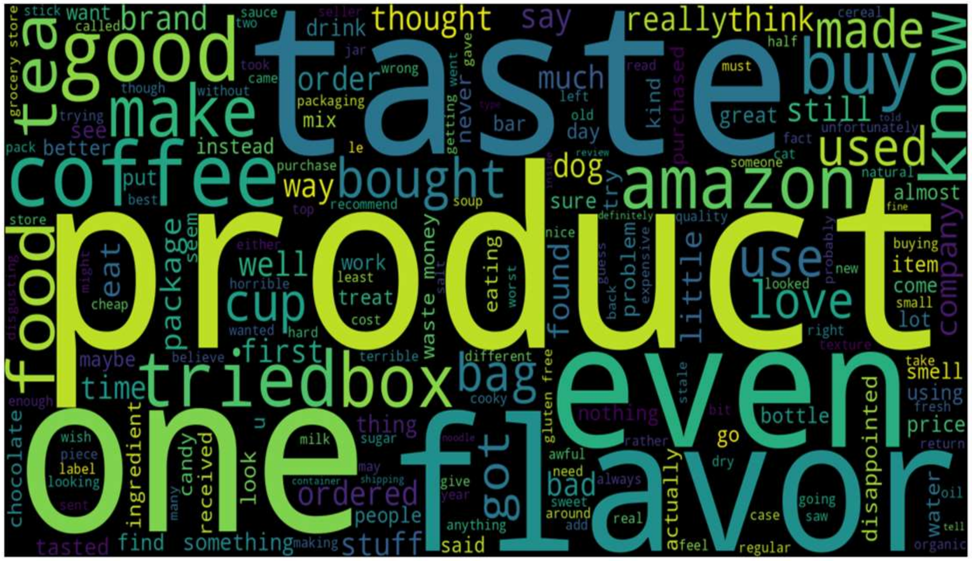


From the above we can observe the user IDs of the Top 15 Customers who are most active and leave more reviews on products.

EDA (Cont.): For ReviewText

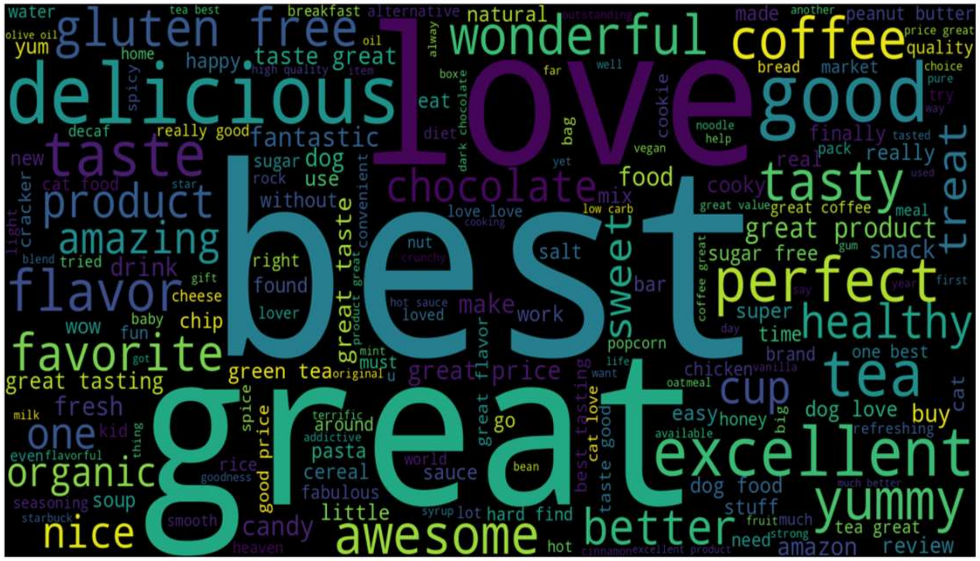


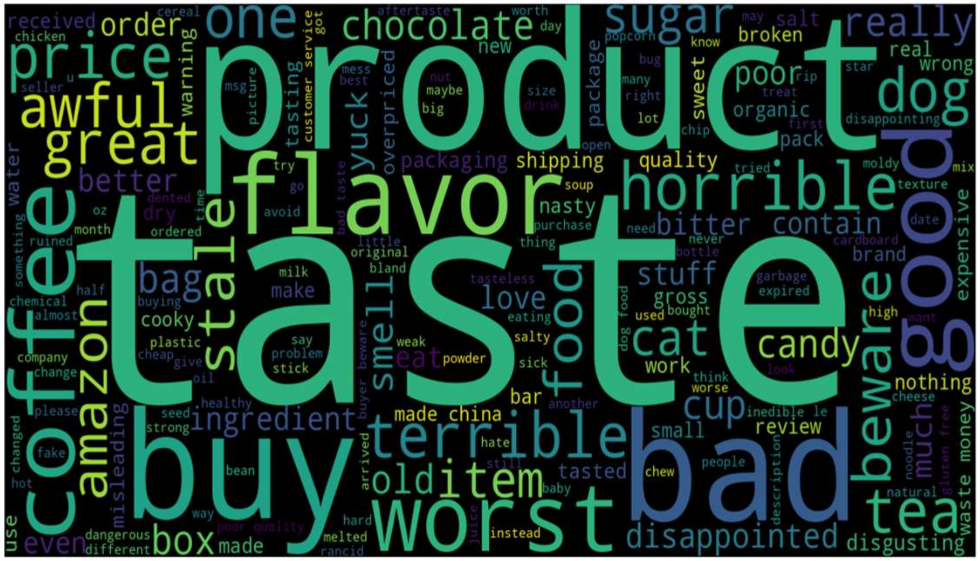
From the above we can observe that the 5 score reviews have positive words like great product , highly recommend, wonderful, etc. which shows that the customers are satisfied with this products.



From the above we can observe that the 1 score reviews have negative words like bad, awful, terrible. worst, etc. which signifies that the customers are not satisfied with certain products.

EDA (Cont.): For ReviewSummary





From the above we can observe that the 5 score reviews have positive words like excellent , great, favorite, best, love etc. which shows that the customers are satisfied with this products.

From the above we can observe that the score 1 reviews have negative words like disappointed, horrible, disgusting. gross, etc. which signifies that the customers are not satisfied with certain products.

Model Building

For the classification task of score prediction, the below machine learning models were used:

- **K-Nearest Neighbors (KNN) Classifier:** A non-parametric method that classifies data points based on the majority class among their k-nearest neighbors, which can be effective for capturing local patterns in the data.
- **Logistic Regression:** A classic binary classification algorithm that models the probability of a customer churning based on input features. It's interpretable and efficient for linearly separable data.
- **Decision Trees:** These models partition the feature space into regions and make predictions based on majority voting within each region. They're intuitive and can capture non-linear relationships between features and the target variable.
- **Random Forest:** An ensemble method that builds multiple decision trees and combines their predictions to improve accuracy and robustness. It's effective for handling complex datasets and mitigating overfitting.
- **Support Vector Machines (SVM):** SVMs aim to find the hyperplane that best separates the data points into different classes. They're effective in high-dimensional spaces and suitable for datasets with clear margins of separation.
- **XGBoost Classifier:** XGBoost, short for eXtreme Gradient Boosting, is a popular machine learning algorithm known for its efficiency and performance in supervised learning tasks, particularly in structured/tabular data and gradient boosting frameworks.
- **Naive Bayes Classifier**: The Naïve Bayes classifier is a probabilistic machine learning algorithm based on Bayes' Theorem with an assumption of independence between features. Despite its simplicity and "naïve" assumption, it's surprisingly effective in many-real world applications, especially in text classification and sentiment analysis.

Model Training

The model training process involved:

- Splitting the dataset into training and testing sets (85:15 Ratio) to train the models on a subset of the data and evaluate their performance on unseen data.
- Training each model using the training set, where the model learns patterns and relationships between features and the target variable (score).

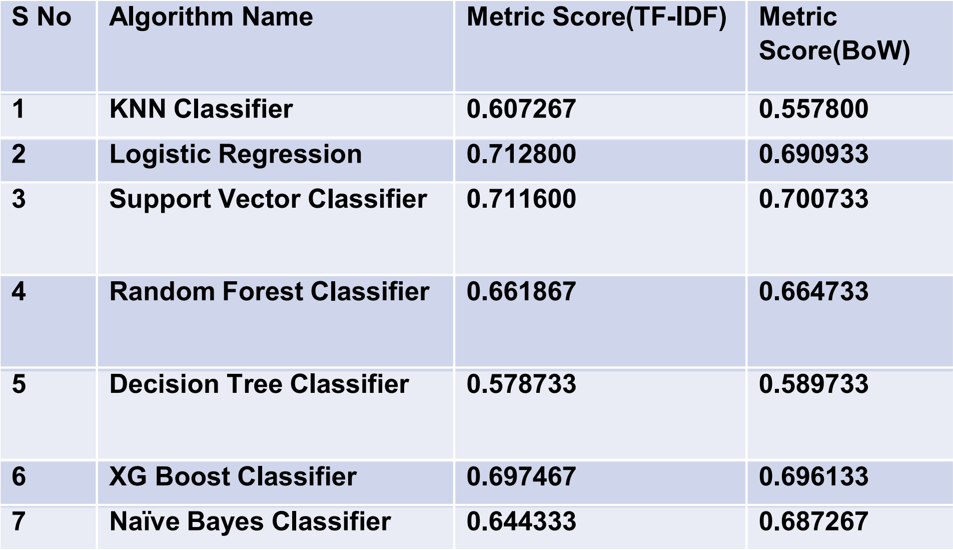
S No	Type of Problem	Algorithm Name
1	Classification	KNNeighbors Classifier
2	Classification	Logistic Regression
3	Classification	SVC
4	Classification	Random Forest Classifier
5	Classification	Decision Tree Classifier
6	Classification	XG Boost Classifier
7	Classification	Naïve Bayes Classifier

Model Evaluation

Evaluation metrics employed to assess model effectiveness was accuracy:

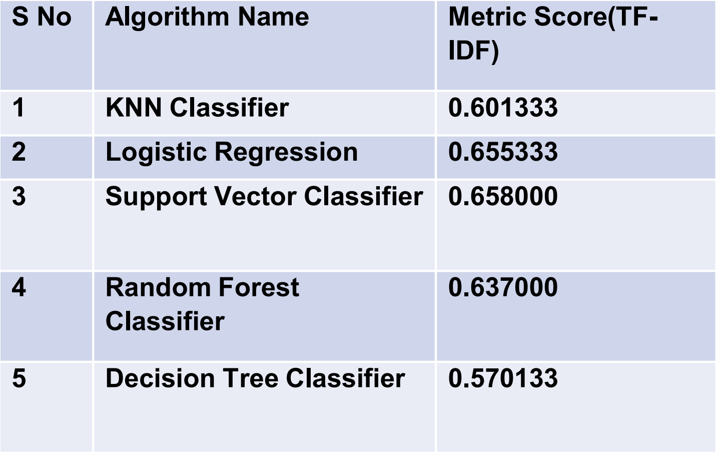
- **Accuracy:** The accuracy metric measures the proportion of correctly classified instances out of total instances, providing a straightforward evaluation of overall model performance. It is commonly used in classification tasks to assess the model's ability to correctly predict the class labels of the dataset.

For ReviewText:



S No	Algorithm Name	Metric Score(TF-IDF)	Metric Score(BoW)
1	KNN Classifier	0.607267	0.557800
2	Logistic Regression	0.712800	0.690933
3	Support Vector Classifier	0.711600	0.700733
4	Random Forest Classifier	0.661867	0.664733
5	Decision Tree Classifier	0.578733	0.589733
6	XG Boost Classifier	0.697467	0.696133
7	Naïve Bayes Classifier	0.644333	0.687267

For ReviewSummary:



S No	Algorithm Name	Metric Score(TF-IDF)
1	KNN Classifier	0.601333
2	Logistic Regression	0.655333
3	Support Vector Classifier	0.658000
4	Random Forest Classifier	0.637000
5	Decision Tree Classifier	0.570133

Challenges Faced

- While identifying the missing values inside the reviewtext and reviewsummary columns, we've also observed that there were many broken html tags which didn't carry any meaning. We have also faced issues while training the models since some models took more time.

Conclusion

- From the above Accuracy results we can observe that Logistic Regressor, Support Vector Classifier and XGBoost Classifier has the highest accuracy Score when compared to all the other models i.e. they have more than 0.69 accuracy score.
- And TF-IDF and BoW gives nearly the same accuracy score for all the models except for Naïve Bayes, for which BoW has higher accuracy and KNN Classifier, for which TF-IDF has better accuracy score.
- And ReviewText gave a better result compared to ReviewSummary.

THANK YOU

INNOMATICS
RESEARCH LABS