Sentiment Analysis : using NLP for E-commerce data of Apparel reviews

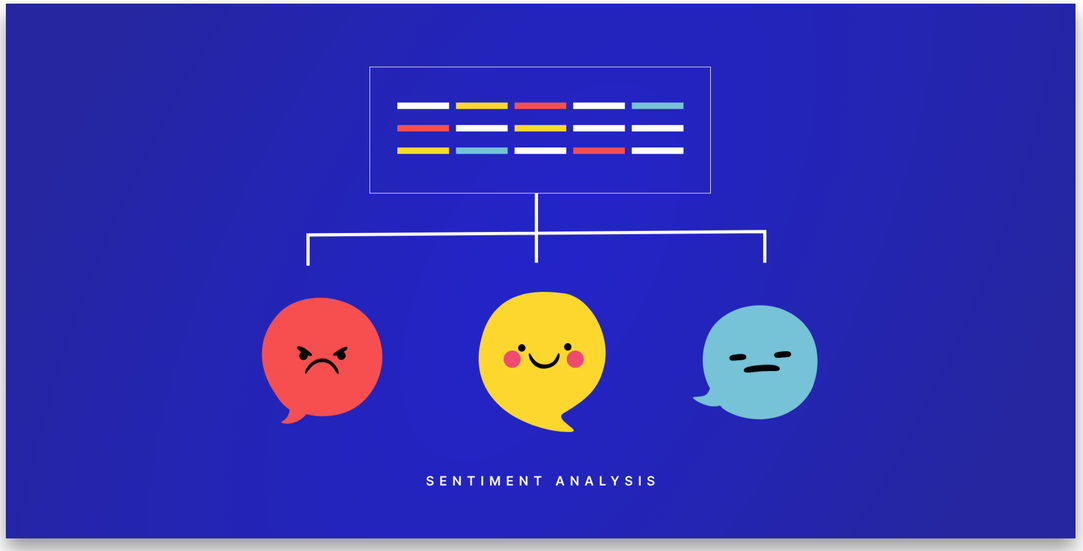
Data Science Diploma Program
Capstone - Sprint2

Presented By:-
Chandana Chaudhry

Introduction

Problem at hand : Understanding customer sentiments is of paramount importance in marketing strategies and product improvement and figuring out a way to use qualitative data quantitatively.

Sentiment Analysis : is the process of analyzing digital text to determine if the emotional tone of the message is positive, negative, or neutral.



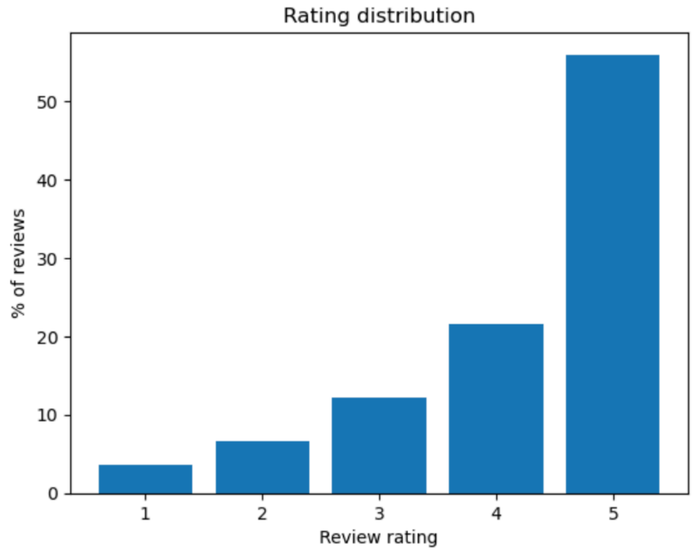
Our Approach/Objective : Sentiment Analysis using NLP with the help of using e-commerce data of apparel reviews.

References : *https://aws.amazon.com/what-is/sentiment-analysis/*
Image src: *https://exemplary.ai/blog/sentiment-analysis*

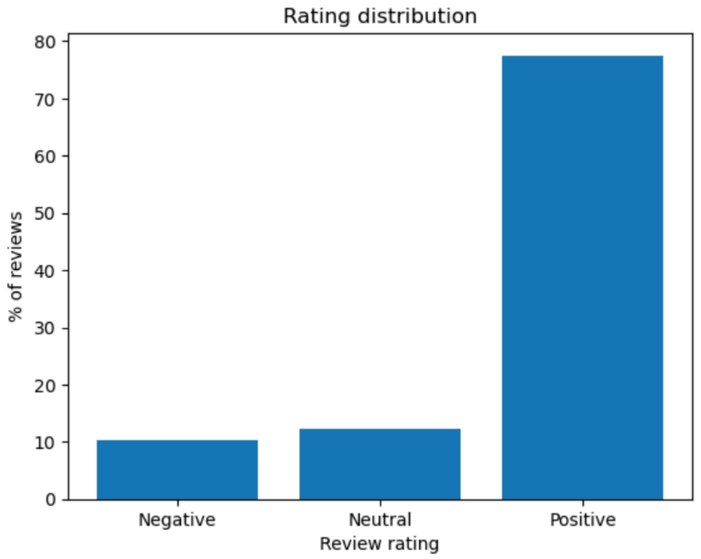
Exploratory Data Analysis and Insights

- Shape of the Data Set is 23486 rows and 11 columns
- 5 Object type columns and 6 int types
- The data set contains ‘Title : 3810’ and ‘Review_text : 845’ null values
- Classification problem with ‘Rating’ as the target variable
- After plotting a heat map, there is high correlation between ‘Rating’ and ‘Recommendation_IND’.
- The data at hand had to be processed into three classification : ‘Positive’ , ‘Neutral’ and ‘Negative’. Initially it had 5 classes
- Also, the data at hand was imbalanced.

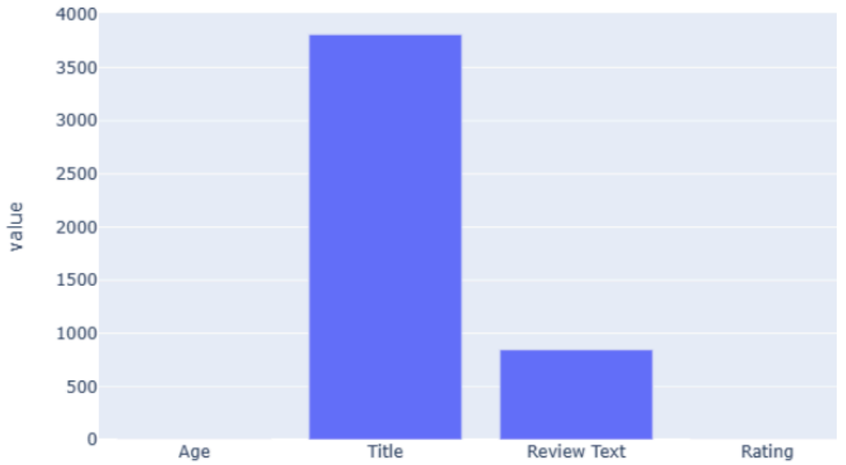
Exploratory Data Analysis and Insights



The rating distribution before processing



The rating distribution after processing



Null value distribution

Data Preprocessing :

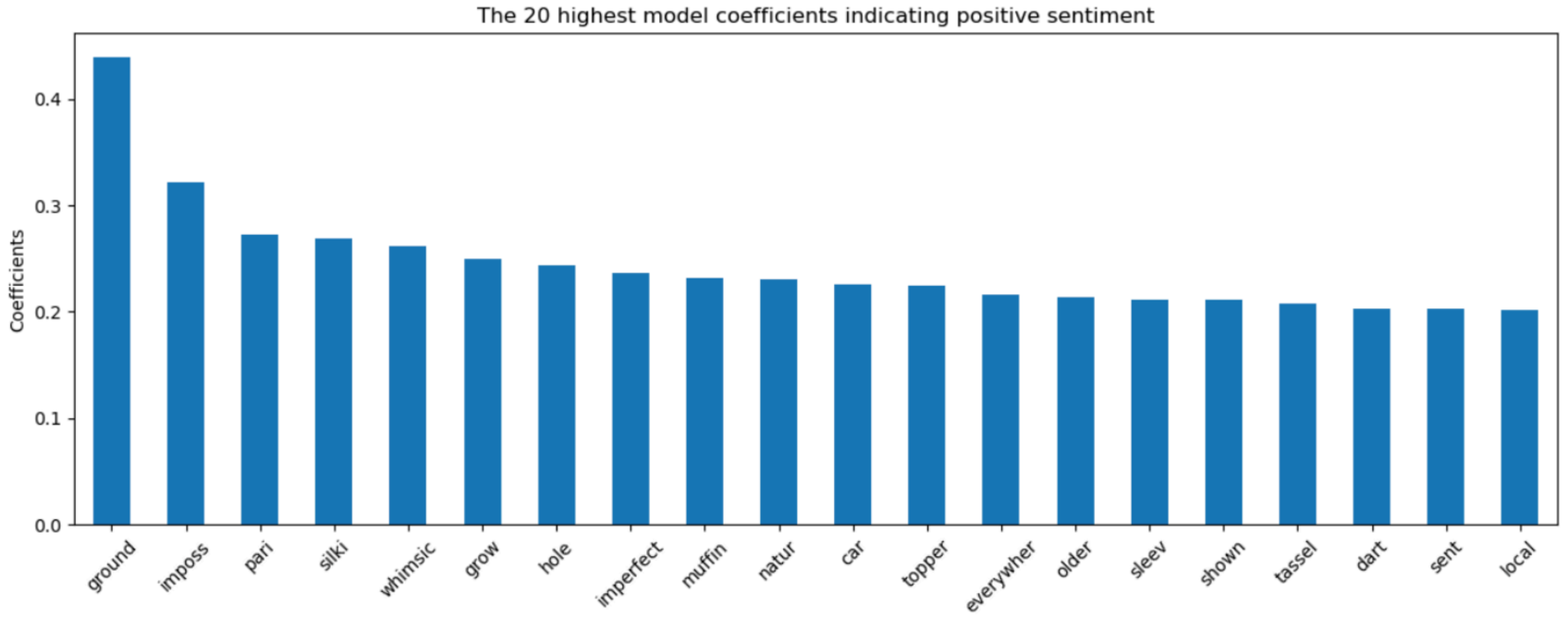
- Dropping the Title column, as it is not relevant to the task of predicting the rating of a review given the text of the review.
- Removing any rows that are missing the rating label.
- Binarizing the rating column into three categories: Negative, Neutral, and Positive. Dropping the original rating column, as it is no longer needed.
- Removing any rows that are missing any values after the previous steps.
- Tokenizing data using NLP BagofWords technique.
- Concatenating ‘Tokens’ to our data frame before model fit step.

Baseline Model or Model 1 : Logistic Regression

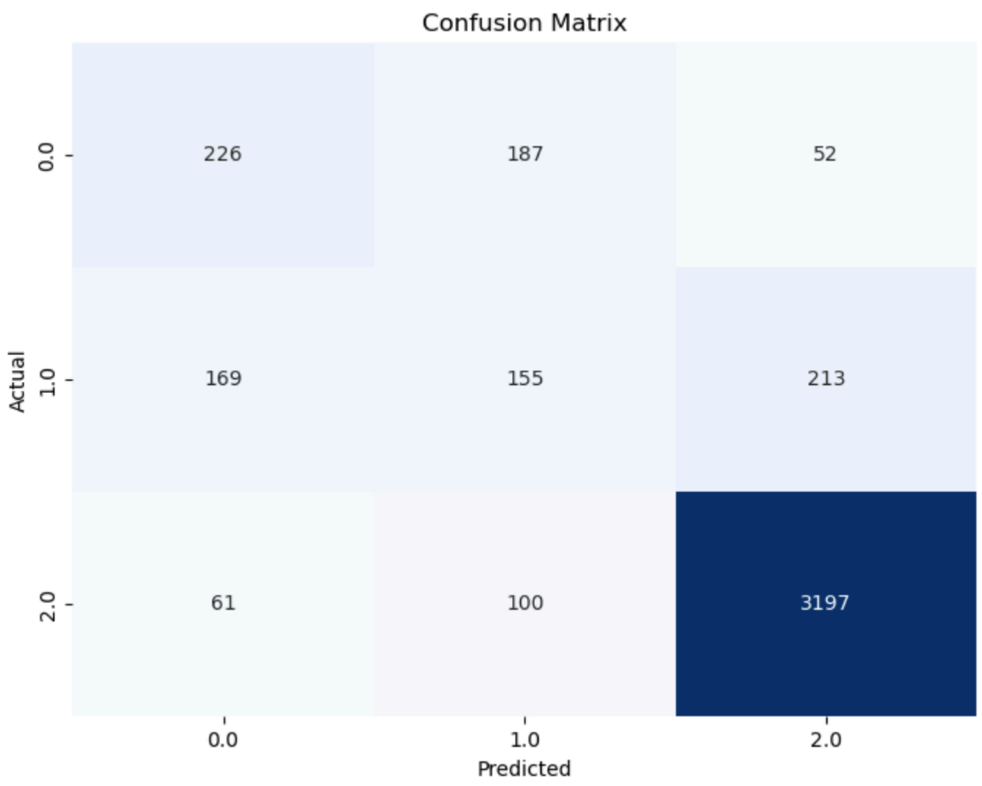
Logistic regression is one of the most basic (yet effective) tools we have for classifying categorical data.

We plotted the top 20 tokens with highest Log.coefficients

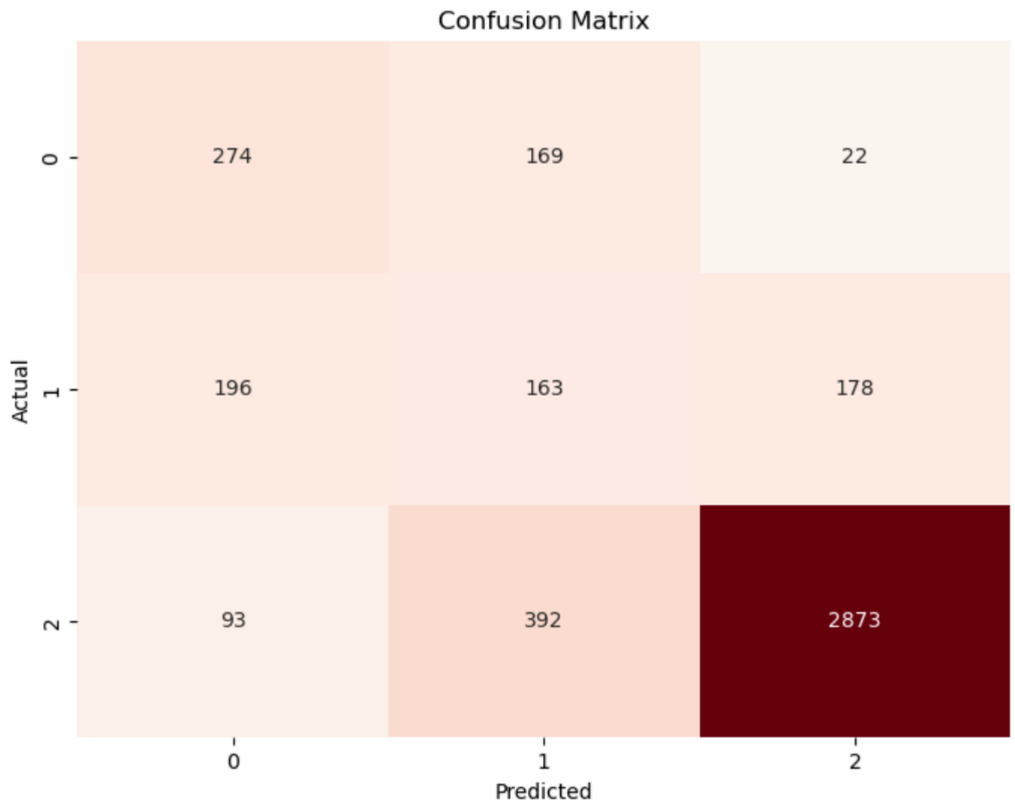
Unbalanced Data :
Logistic regression
Accuracy Score : 82%



Logistic Regression : Balanced vs Unbalanced data



Log. Regression W/O Data Balancing



Log. Regression with Data Balancing

Comparison : Balanced vs Unbalanced data

Logistic Regression(w/o balance)

Class 0 (negative)

- Precision: 0.50
- Recall: 0.49
- F1-score: 0.49

Class 1 (neutral)

- Precision: 0.35
- Recall: 0.29
- F1-score: 0.32

Class 2 (positive)

- Precision: 0.92
- Recall: 0.95
- F1-score: 0.94

Accuracy: 0.82

Logistic Regression(with balance)

Class 0 (negative)

- Precision: 0.49
- Recall: 0.59
- F1-score: 0.53

Class 1 (neutral)

- Precision: 0.23
- Recall: 0.30
- F1-score: 0.26

Class 2 (positive)

- Precision: 0.93
- Recall: 0.86
- F1-score: 0.89

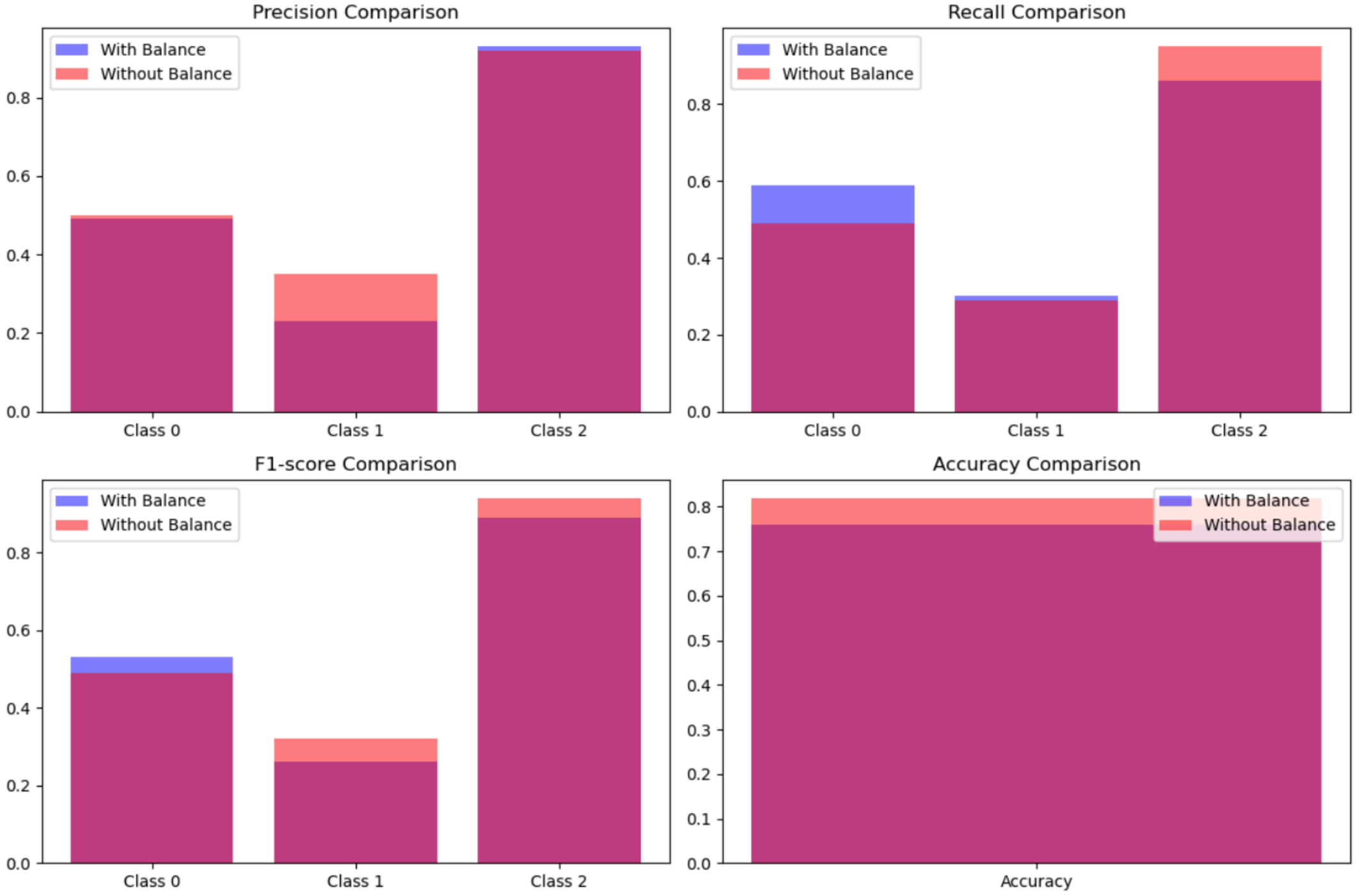
Accuracy: 0.76

Classification Report:

	precision	recall	f1-score	support
0.0	0.50	0.49	0.49	465
1.0	0.35	0.29	0.32	537
2.0	0.92	0.95	0.94	3358
accuracy			0.82	4360
macro avg	0.59	0.58	0.58	4360
weighted avg	0.81	0.82	0.81	4360

Classification Report:

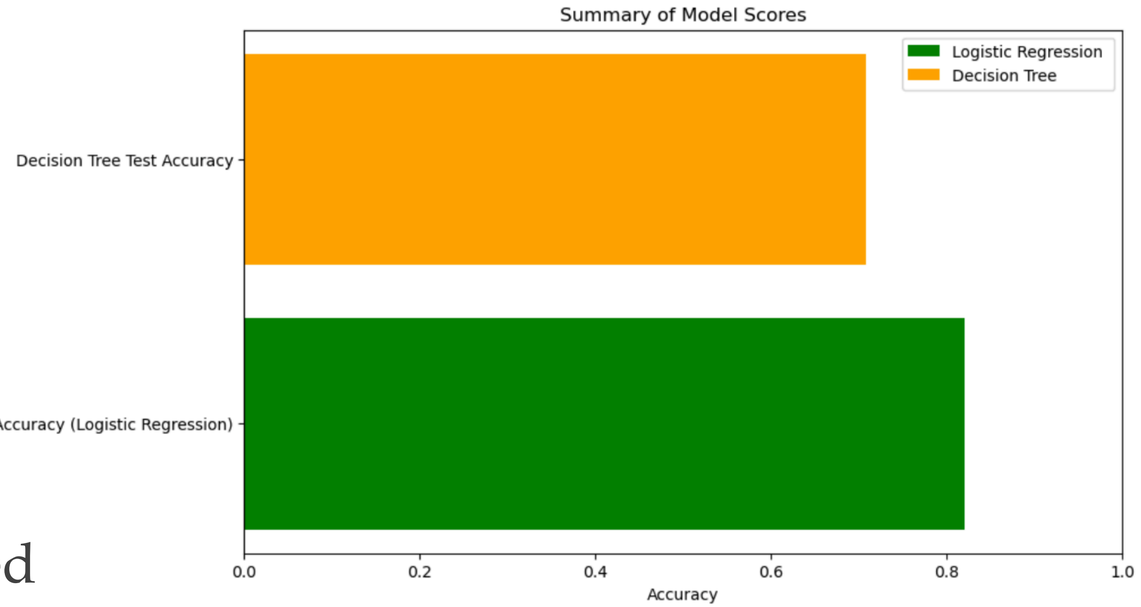
	precision	recall	f1-score	support
0.0	0.49	0.59	0.53	465
1.0	0.23	0.30	0.26	537
2.0	0.93	0.86	0.89	3358
accuracy			0.76	4360
macro avg	0.55	0.58	0.56	4360
weighted avg	0.80	0.76	0.78	4360


Precision Comparison
With Balance
Without Balance
0.0
0.2
0.4
0.6
0.8
Class 0
Class 1
Class 2
Recall Comparison
With Balance
Without Balance
0.0
0.2
0.4
0.6
0.8
Class 0
Class 1
Class 2
F1-score Comparison
With Balance
Without Balance
0.0
0.2
0.4
0.6
0.8
Class 0
Class 1
Class 2
Accuracy Comparison
With Balance
Without Balance
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Accuracy

Model 2 : Decision Trees

- Performed using pipeline and with PCA (n_components = 0.90) preserving 90% of variance
- Max_depth = 10
- Min_sample leaf = 1
- Model Accuracy = 70%

Concluding that Logistic Regression performed better



Next Iterations

- Deep Dive and troubleshoot RNN LSTM neural network with TensorFlow using Self Attention layer
- Exploring Recommender Systems : which essentially helps us create a user filtering as per user needs. The filtering is done on the basis of cosine similarity.
- Adding interactive visualizations

Thank you and References

- https://www.kaggle.com/datasets/nicapotato/womens-ecommerce-clothing-reviews
- **Sentiment Classification :**https://www.researchgate.net/publication/323545316_Statistical_Analysis_on_E-Commerce_Reviews_with_Sentiment_Classification_using_Bidirectional_Recurrent_Neural_Network , **paper pdf link : https://arxiv.org/pdf/1805.03687.pdf**
- https://aws.amazon.com/what-is/sentiment-analysis/
- https://medium.com/@zaiinn440/attention-is-all-you-need-the-core-idea-of-the-transformer-bbfa9a749937
- **Attention is all you need :** https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- **Case Analysis - Twitter** : https://www.academia.edu/31874952/Twitter_Sentiment_Analysis_