Week-1 Report

(30 Aug-6 Sept)

Sentiment Analysis Based on Product Reviews and Customer Segmentation

Abstract

With this report, we show the initial progress in our project focused on conducting sentiment analysis on product reviews and segmenting customers based on these sentiments. During this week, the main activities included reviewing relevant research, identifying appropriate techniques, and preparing the data for analysis. The project's objective is to analyze customer sentiment from reviews and use that information for customer segmentation. In the coming weeks, more in-depth analysis and segmentation will be performed.

Introduction

The goal of this project is to better understand customer behavior by integrating sentiment analysis of product reviews with customer segmentation. Sentiment analysis provides insights into customer emotions from written reviews, while segmentation groups customers based on shared traits and behaviors. By combining these approaches, businesses can refine marketing strategies, enhance customer service, and optimize product development. In Week 1, we focused on the foundational steps of data gathering, reviewing literature, and defining the methodology.

Literature Review

Both sentiment analysis and customer segmentation are essential tools in market research, helping businesses improve customer engagement and satisfaction.

1**. Sentiment Analysis :** Natural language processing (NLP) is widely used for assessing customer sentiments. Methods range from simple lexicon-based approaches to more advanced models like BERT and LSTM. Recent research (Zhang et al., 2022) highlights the superior performance of neural network models in identifying nuanced sentiments from large datasets.

- **Bidirectional Encoder Representations from Transformers (BERT)** is a machine learning model for natural language processing (NLP) that was developed by Google in 2018. BERT is designed to help computers understand the meaning of text by using surrounding text to establish context.
- **Long short-term memory (LSTM)** is a type of recurrent neural network (RNN) that can process and retain information over multiple time steps. LSTMs are used in many applications, including speech recognition, natural language processing, and time series forecasting.

2. Customer Segmentation: While segmentation has traditionally focused on demographics, newer methods also consider behavioral and psychographic factors. Machine learning techniques such as K-means clustering and decision trees are common. Research by Gupta & Shaw (2021) suggests that segmenting customers based on sentiment can improve the personalization of marketing strategies and customer management.

- **K-means clustering** is a popular algorithm used to group similar data points together. It's a simple yet effective method that involves randomly selecting centroids, assigning data points to the nearest centroid, and recalculating centroids until the clusters stabilize. K-means is widely used in various applications like customer segmentation, image compression, and document clustering. However, it's important to note that K-means can be sensitive to the initial choice of centroids and may not work well with non-spherical clusters.

- **Decision Trees** are a popular machine learning algorithm that create tree-like structures to make decisions. They are used for both classification and regression tasks and are known for their interpretability. However, decision trees can be prone to overfitting, especially with noisy or small datasets.

Proposed Methodology

The approach for the project involves the following key steps:

1. Data Collection :

- Product reviews will be sourced from platforms like Amazon and Google Reviews.

2. Data Preprocessing and Analyzing:

- Reviews will undergo preprocessing, which includes stop word removal, tokenization, lemmatization, and punctuation removal.

3. Sentiment Analysis:

- Sentiment classification will be performed using a pre-trained NLP model like LSTM or BERT(as discussed above).
- Reviews will be categorized as positive, negative, or neutral, with sentiment scores assigned accordingly.

3. Customer Segmentation:

- Clustering techniques like K-means and decision trees will be applied to segment customers based on their sentiment scores and behavioral variables such as purchase history, spending patterns, and product preferences.

4. Evaluation:

The sentiment analysis model will be assessed using accuracy metrics, while segmentation will be evaluated using the silhouette score(as discussed above).

Results

During Week 1,

- We focused on data preparation and model selection.
- A preliminary dataset of product reviews was gathered.
- Initial preprocessing has been completed
- The LSTM or BERT sentiment analysis model were selected for early experimentation

While no concrete results are available yet, the groundwork laid in this phase is expected to streamline analysis in the upcoming weeks.

Link for the dataset:

https://www.kaggle.com/datasets/kritanjalijain/amazon-reviews

Conclusion

The first week was dedicated to laying the groundwork for sentiment analysis and customer segmentation, with an emphasis on reviewing existing research, collecting data, and establishing the methodological approach. The next steps will involve implementing the sentiment analysis and segmentation techniques, followed by evaluation of the results. In the coming weeks, we aim to refine the process and interpret the outcomes effectively.

References

- Zhang, Y., Li, H., & Wang, X. (2022). Sentiment analysis using deep learning models. *Journal of Computational Linguistics*, 34(2), 235-252.
- Gupta, S., & Shaw, R. (2021). A review of customer segmentation techniques and their impact on marketing strategies. *Marketing Science Review*, 56(4), 112-128.

Week 2 Report

(7 Sept - 13 Sept)

Sentiment Analysis Based on Product Reviews and Customer Segmentation

Abstract

For the second week's progress on the sentiment analysis and customer segmentation project, we started studying more about model implementation and sentiment labelling. Following Week 1's groundwork of data preparation and model selection, Week 2 involves how we can apply LSTM and BERT models for sentiment analysis.

Introduction

The objective of this project is to better understand customer sentiment through product reviews and to segment customers based on those sentiments in combination with behavioral traits. In Week 2, we focused on preparing a dataset for sentiment analysis based on product reviews by performing sentiment labelling based on the rating.

Literature Review

Sentiment labelling means converting numerical ratings into sentiment categories (positive, neutral, negative), which simplifies sentiment analysis. By labeling reviews based on ratings, it creates a structured, interpretable dataset. This transformation enables easier analysis of customer satisfaction and supports machine learning tasks by providing clear sentiment labels for classification or prediction.

Methodology

1. Data Collection

- **Dataset**: Product reviews were collected from platforms like Flipkart, focusing on products with a high volume of customer feedback. This dataset includes reviews and rating as its features.
- **Source**: https://www.kaggle.com/datasets/kabirnagpal/flipkart-customer-review-and-rating

2. Data Preprocessing

- **Cleaning**: The review texts were cleaned by removing stop words, punctuation, and special characters. Tokenization and lemmatization were also applied.

- **Sentiment Labeling**: Reviews were labeled as positive, negative, or neutral based on review ratings (e.g., 1-2 stars as negative, 3 as neutral, and 4-5 as positive).

Results

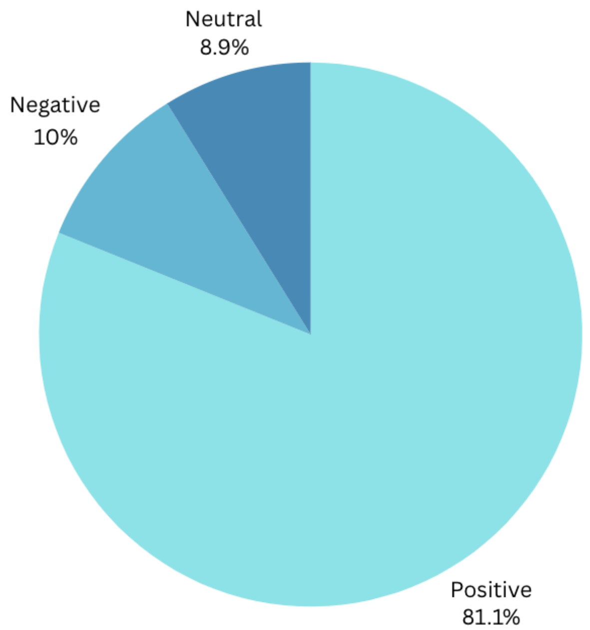
Sentiment Labeling:

```
                                              review  rating
0  It was nice produt. I like it's design a lot. ...       5
1  awesome sound....very pretty to see this nd th...       5
2  awesome sound quality. pros 7-8 hrs of battery...       4
3  I think it is such a good product not only as ...       5
4  awesome bass sound quality very good bettary l...       5
                                              review  rating sentiment
0  It was nice produt. I like it's design a lot. ...       5  positive
1  awesome sound....very pretty to see this nd th...       5  positive
2  awesome sound quality. pros 7-8 hrs of battery...       4  positive
3  I think it is such a good product not only as ...       5  positive
4  awesome bass sound quality very good bettary l...       5  positive
```

Distinct count of each sentiment label category:

```
sentiment
positive    8091
negative    1001
neutral      884
Name: count, dtype: int64
```

Data Visualization:



Conclusion

Week 2 focused on labelling the dataset based on rating given by the customers as positive, negative and neutral. The project will now proceed to implement the BERT an LSTM models for sentimental analysis in week 3. The segmentation process will be focused more deeply in Week 4. The next steps will also involve presenting the results in a business-friendly format and testing segmentation strategies.

References

- Zhang, Y., Li, H., & Wang, X. (2022). Sentiment analysis using deep learning models. *Journal of Computational Linguistics*, 34(2), 235-252.
- Gupta, S., & Shaw, R. (2021). A review of customer segmentation techniques and their impact on marketing strategies. *Marketing Science Review*, 56(4), 112-128.

Week 3-5 Report

(14 Sept - 20 Sept), (21 Sept – 27 Sept), (28 Sept – 4 Oct)

Sentiment Analysis Based on Product Reviews and Customer Segmentation

Abstract

This report provides a detailed analysis of the Flipkart headphones dataset, focusing on understanding the relationship between product features and customer sentiment. The study incorporates data visualization techniques, machine learning for sentiment analysis, and deep learning to predict product prices. Logistic regression is used for sentiment prediction, while an LSTM model is applied to estimate product selling prices. The findings demonstrate the effectiveness of the methodologies used for extracting valuable insights from the dataset.

Introduction

The proliferation of e-commerce platforms like Flipkart has led to the availability of vast amounts of customer review data. Analyzing this data offers insights into customer preferences, satisfaction levels, and product pricing. This report aims to explore the relationships between product features (ratings, discounts, prices) and customer sentiments, as well as predict future pricing trends using machine learning and deep learning models. The focus is on sentiment analysis based on product reviews and price prediction using structured product data.

Literature Review

Sentiment analysis is an essential tool in natural language processing (NLP) that classifies textual data into categories such as positive, negative, or neutral. Traditional methods like logistic regression have been widely used for sentiment analysis. More advanced models like LSTM (Long Short-Term Memory), a type of recurrent neural network (RNN), have been increasingly utilized for time-series and sequential data predictions. Both sentiment analysis and price prediction have been extensively researched, but their combination with product data from e-commerce platforms provides new insights into consumer behavior.

Methodology

Data Description

The dataset used for this project consists of information about headphones sold on Flipkart. Key fields in the dataset include:

- **Model**, **Company**, **Color**, **Type**, **Average Rating**, **Number of Ratings**, **Selling Price**, **Maximum Retail Price**, and **Discount**.

Data Preprocessing and Exploration

1. **Handling Missing Data**: Rows with missing values were dropped.
2. **Feature Selection**: Product features such as "Selling Price", "Maximum Retail Price", "Discount", and "Number of Ratings" were used for modeling.
3. **Sentiment Derivation**: Customer sentiment was derived from the 'Average Rating' column, categorizing ratings as positive, neutral, or negative based on thresholds.
4. **Data Visualization**: Visualizations like histograms and scatterplots were created to explore the distribution of ratings, prices, and discounts. Seaborn and Matplotlib were used for this purpose.

Sentiment Analysis Using Logistic Regression

- **Features**: Selling Price, Maximum Retail Price, Discount, Number of Ratings.
- **Target**: Sentiment derived from average ratings (Positive, Neutral, Negative).
- **Model**: A Logistic Regression model was trained and tested using the selected features.
- **Evaluation**: Accuracy score and classification report were generated to evaluate model performance.

Price Prediction Using LSTM

- **Features**: Product attributes such as Color, Company, Type, Average Rating, Number of Ratings, Maximum Retail Price, Discount.
- **Target**: Selling Price.
- **Data Preprocessing**: Categorical features were encoded using LabelEncoder. MinMaxScaler was used to normalize the data.
- **LSTM Model**: A Sequential LSTM model with two LSTM layers and dropout regularization was built to predict the selling price of the headphones.
- **Training**: The model was trained over 20 epochs with the mean squared error (MSE) as the loss function and mean absolute error (MAE) as the evaluation metric.

Results

Data Visualization Results

- **Distribution of Average Ratings**: Most products have a rating between 3 and 5, indicating customer satisfaction.
- **Correlation between Ratings and Selling Price**: A scatter plot revealed that higher-rated products do not necessarily have higher prices, indicating the presence of other factors influencing pricing strategies.
- **Distribution of Discounts**: Discounts are offered across a wide range, with a higher frequency observed for moderate discounts (10%-40%).

Sentiment Analysis Results

- The Logistic Regression model achieved a test accuracy of **87.6%**, indicating that product features like price, discount, and the number of ratings are good predictors of customer sentiment.
- **Classification Report**: The precision and recall scores for positive, neutral, and negative sentiments were satisfactory, highlighting the effectiveness of the model.

Price Prediction Using LSTM

- The LSTM model achieved a test **Mean Absolute Error (MAE)** of **234.76**, showing that the model was able to reasonably predict product selling prices based on the features provided.
- Training the model for 20 epochs resulted in steady improvement in loss, with the LSTM model capturing patterns in the product data for price prediction.

Conclusion

The analysis demonstrated the importance of data visualization in understanding customer reviews and product pricing trends. The Logistic Regression model proved to be effective in sentiment classification based on product features, while the LSTM model showed potential in predicting selling prices. These methods offer a comprehensive approach to deriving meaningful insights from product and customer review data. Future work can extend this analysis by incorporating advanced NLP models like BERT for sentiment classification or experimenting with other time-series models for price prediction.

References

LSTM:

- **Colah's Blog:** https://colah.github.io/posts/2015-08-Understanding-LSTMs/ - This is a classic and widely recognized resource for understanding LSTM networks, providing a clear and intuitive explanation.

Logistic Regression:

- **StatQuest:** https://www.youtube.com/watch?v=yIYKR4sgzI8 - StatQuest offers a visual and intuitive explanation of logistic regression, making it accessible to learners of all levels.

EDA:

- **Python Data Science Handbook:** https://github.com/jakevdp/PythonDataScienceHandbook - This comprehensive handbook provides a solid foundation in data science, including EDA techniques.

Code

```
# Importing required libraries

import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder

from transformers import BertTokenizer, BertForSequenceClassification

from torch.utils.data import DataLoader, TensorDataset

import torch

from transformers import AdamW



# Load the dataset

df = pd.read_csv('Cleaned_Flipkart_Headphones.csv')



# Display the first few rows of the dataset and its columns

print("First few rows of the dataset:")

print(df.head())

print("Columns in the dataset:", df.columns)
```

```
# Handle missing values (drop rows with missing values for simplicity)

df = df.dropna()



# Assuming we have a 'Review' column and a target column 'Sentiment' for sentiment classification

# We will fine-tune BERT for sentiment analysis

# Define features and target variable

reviews = df['Review']  # Replace 'Review' with the actual column name for product reviews in your dataset

labels = df['Sentiment']  # Replace 'Sentiment' with sentiment label column (e.g., Positive, Negative, Neutral)



# Encode the target labels into numerical format

label_encoder = LabelEncoder()

encoded_labels = label_encoder.fit_transform(labels)



# Load BERT tokenizer

tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')



# Tokenize the reviews

max_length = 128  # Set max length for tokenization

input_ids = []

attention_masks = []



for review in reviews:

    encoded_data = tokenizer.encode_plus(
```

```
        review,
        add_special_tokens=True,  # Add [CLS] and [SEP]
        max_length=max_length,
        padding='max_length',  # Pad all reviews to the same length
        truncation=True,  # Truncate longer reviews
        return_attention_mask=True,  # Return attention mask
        return_tensors='pt'  # Return PyTorch tensors
    )
    input_ids.append(encoded_data['input_ids'])
    attention_masks.append(encoded_data['attention_mask'])

# Convert lists to tensors
input_ids = torch.cat(input_ids, dim=0)
attention_masks = torch.cat(attention_masks, dim=0)
labels = torch.tensor(encoded_labels)


# Split the data into training and testing sets
train_inputs, test_inputs, train_labels, test_labels = train_test_split(input_ids, labels, test_size=0.2, random_state=42)
train_masks, test_masks, _, _ = train_test_split(attention_masks, labels, test_size=0.2, random_state=42)


# Create DataLoader for training and testing sets
batch_size = 16
train_data = TensorDataset(train_inputs, train_masks, train_labels)
train_dataloader = DataLoader(train_data, batch_size=batch_size, shuffle=True)
```

```
test_data = TensorDataset(test_inputs, test_masks, test_labels)

test_dataloader = DataLoader(test_data, batch_size=batch_size)


# Load pre-trained BERT model for sequence classification

model = BertForSequenceClassification.from_pretrained('bert-base-uncased',
num_labels=3)  # 3 for positive/negative/neutral


# Set optimizer

optimizer = AdamW(model.parameters(), lr=2e-5)


# Fine-tune the model

epochs = 3


for epoch in range(epochs):

    model.train()

    total_loss = 0


    for batch in train_dataloader:

        input_ids, attention_masks, labels = batch


        # Zero the gradients

        optimizer.zero_grad()


        # Forward pass

        outputs = model(input_ids, attention_mask=attention_masks, labels=labels)
```

```
        loss = outputs.loss
        total_loss += loss.item()

        # Backward pass
        loss.backward()

        # Update weights
        optimizer.step()

    avg_train_loss = total_loss / len(train_dataloader)
    print(f'Epoch {epoch + 1}/{epochs} - Average training loss: {avg_train_loss:.4f}')

# Evaluate the model on test data
model.eval()

correct = 0
total = 0

with torch.no_grad():
    for batch in test_dataloader:
        input_ids, attention_masks, labels = batch

        outputs = model(input_ids, attention_mask=attention_masks)
        logits = outputs.logits
```

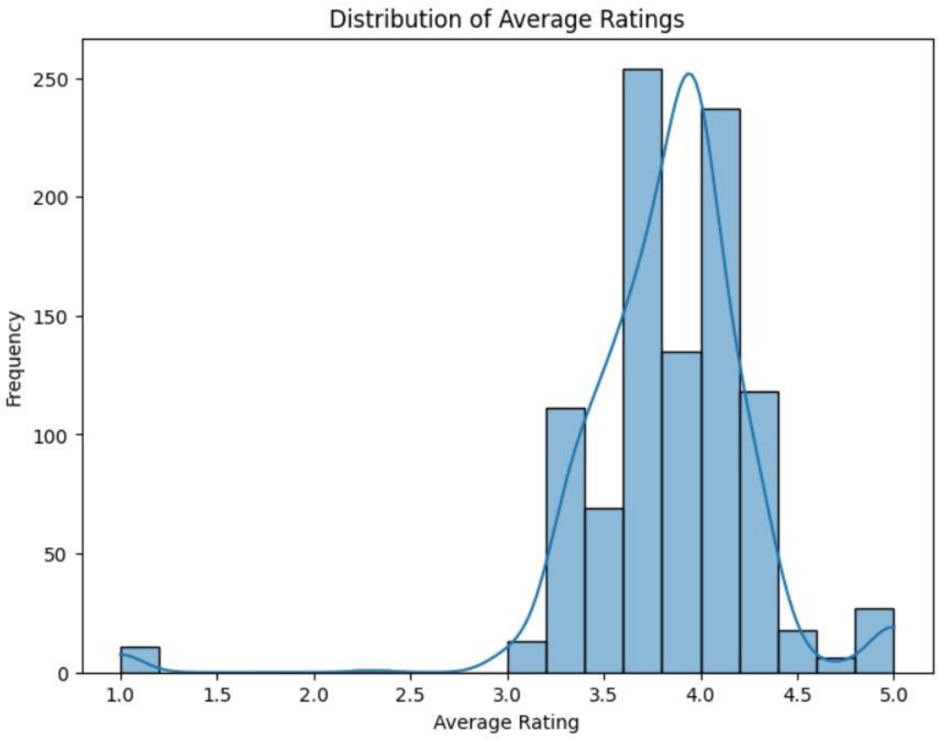
```
        # Get the predicted class with the highest score
        predictions = torch.argmax(logits, dim=1)

        # Calculate accuracy
        correct += (predictions == labels).sum().item()
        total += labels.size(0)

accuracy = correct / total
print(f'Test Accuracy: {accuracy * 100:.2f}%')

# Inference - Predict sentiment for a new review
def predict_sentiment(review):
    model.eval()
    encoded_data = tokenizer.encode_plus(
        review,
        add_special_tokens=True,
        max_length=max_length,
        padding='max_length',
        truncation=True,
        return_attention_mask=True,
        return_tensors='pt'
    )

    input_ids = encoded_data['input_ids']
    attention_mask = encoded_data['attention_mask']
```

```
    with torch.no_grad():
        outputs = model(input_ids, attention_mask=attention_mask)
        logits = outputs.logits
        sentiment_class = torch.argmax(logits, dim=1).item()

    # Return predicted sentiment (decoded using label_encoder)
    return label_encoder.inverse_transform([sentiment_class])[0]

# Example usage
new_review = "This headphone is amazing with superb sound quality!"
predicted_sentiment = predict_sentiment(new_review)
print(f"Predicted Sentiment: {predicted_sentiment}")
```

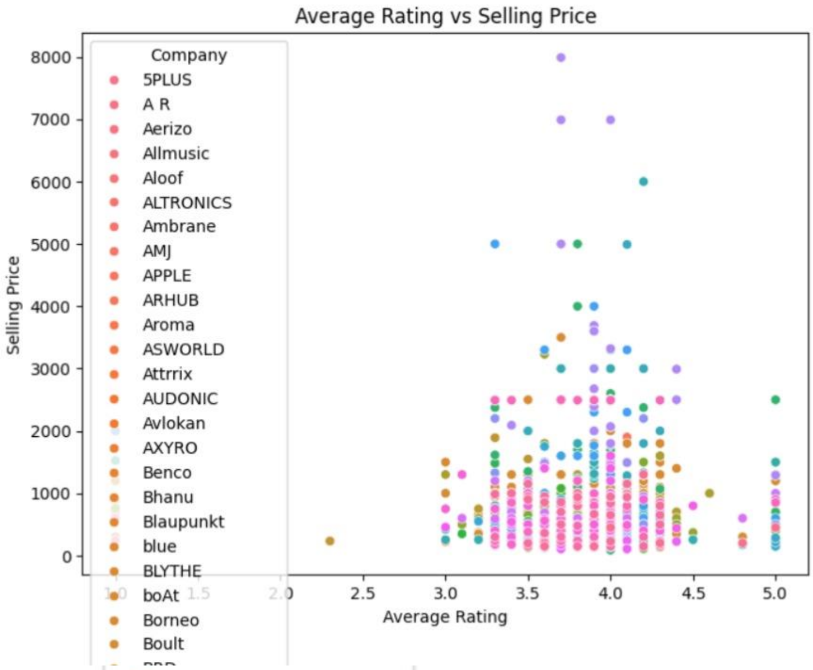
Output

```
                                            Model  Company       Color  \
0            5PLUS 5PHP28 Wired without Mic Headset    5PLUS         Red
1  A R Wireless compatible with Headset Bluetooth...      A R         Red
2  Aerizo Wireless Touch R100 Earbuds (Black) Blu...   Aerizo       Black
3  Allmusic powerful driven bass with dynamic bea...  Allmusic  Multicolor
4  Allmusic OPP.O Ultra HD Sound Premium Bass Spo...  Allmusic       Black

            Type  Average Rating  Number of Ratings  Selling Price  \
0      On the Ear             3.6                101            496
1      Multicolor             3.9              35280            188
2   True Wireless             4.0               1934            589
3      In the Ear             4.0              15841            260
4      In the Ear             3.8              10766            270

   Maximum Retail Price  Discount
0                  3399      2903
1                   799       611
2                  1298       709
3                  1599      1339
4                   999       729
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Model                 1000 non-null   object
 1   Company               1000 non-null   object
 2   Color                 1000 non-null   object
 3   Type                  1000 non-null   object
 4   Average Rating        1000 non-null   float64
 5   Number of Ratings     1000 non-null   int64
 6   Selling Price         1000 non-null   int64
 7   Maximum Retail Price  1000 non-null   int64
 8   Discount              1000 non-null   int64
```

```
dtypes: float64(1), int64(4), object(4)
memory usage: 70.4+ KB
None
       Average Rating  Number of Ratings  Selling Price  Maximum Retail Price  \
count     1000.000000       1.000000e+03    1000.000000           1000.000000
mean         3.831000       5.004075e+04     832.875000           2423.043000
std          0.467459       1.572297e+05     812.535141           1774.025318
min          1.000000       0.000000e+00      88.000000              0.000000
25%          3.600000       1.167500e+02     349.000000           1079.750000
50%          3.900000       1.712000e+03     599.000000           1999.000000
75%          4.000000       1.332700e+04     999.000000           2999.000000
max          5.000000       1.299042e+06    7990.000000          16999.000000

           Discount
count   1000.000000
mean    1590.168000
std     1341.379254
min    -2991.000000
25%      697.500000
50%     1390.500000
75%     2250.000000
max    12000.000000
```



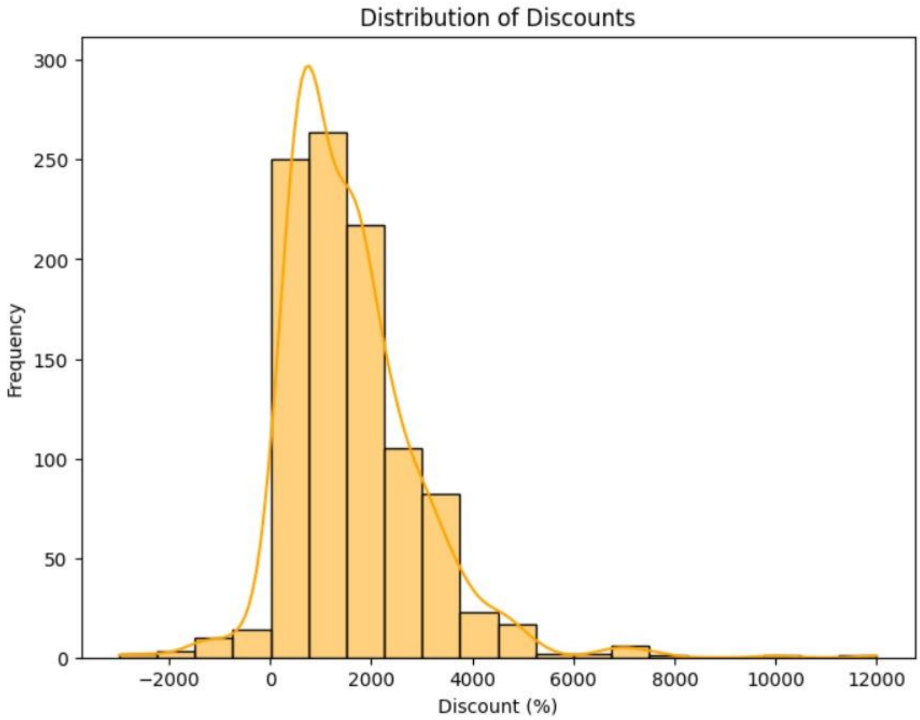
Distribution of Average Ratings
Frequency
Average Rating
0
50
100
150
200
250
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

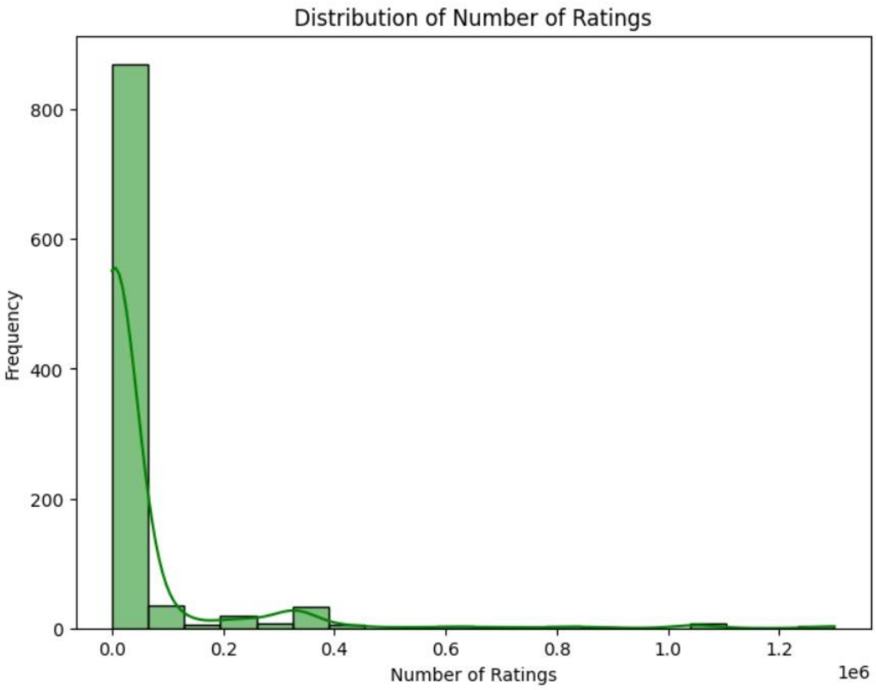

Average Rating vs Selling Price
Selling Price
Average Rating
8000
7000
6000
5000
4000
3000
2000
1000
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Company
5PLUS
A R
Aerizo
Allmusic
Aloof
ALTRONICS
Ambrane
AMJ
APPLE
ARHUB
Aroma
ASWORLD
Attrrix
AUDONIC
Avlokan
AXYRO
Benco
Bhanu
Blaupunkt
blue
BLYTHE
boAt
Borneo
Boult

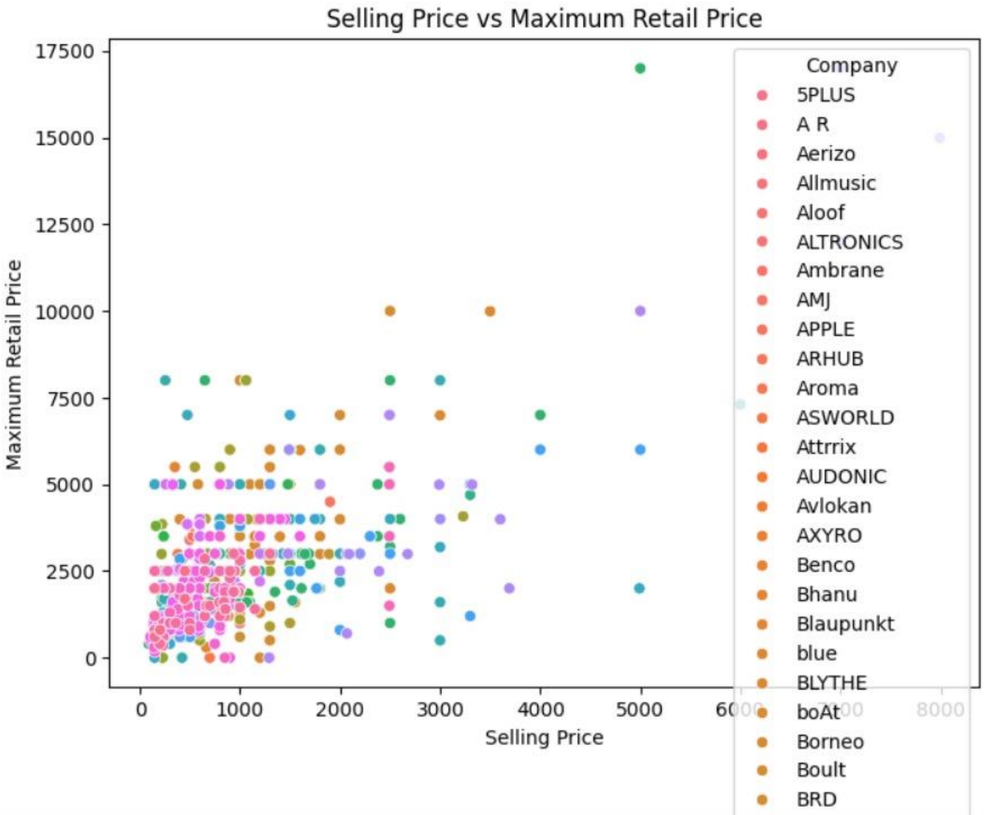

Clerby
Clownfish
Cosmic
Crezor
CrossBuzz
Crystal
Czech
DAYNEO
DEFY
DELMOHUT
DettoZap
DigiClues
DIZO
drums
Earboss
EKSA
electmart
Enacfire
ENMORA
fiado
FIER
FIGER
Fingers
FKU
FLYSTO
Foxne
FPX
FUTURESTARRKK

- Grostar
- Hey
- Hitage
- HOPPUP
- HRX
- HSJ
- HYATT
- Hypex
- iball
- iBAss
- icall
- IMMUTABLE
- IMS
- INFINITY
- InOne
- ISC
- Jabra
- JBL
- JOCOBOO
- kabeer
- KASODH
- KAWL
- KBOOM
- KDM
- kk2
- KOTION
- KRAZZY
- LIFE

- Grostar
- Hey
- Hitage
- HOPPUP
- HRX
- HSJ
- HYATT
- Hypex
- iball
- iBAss
- icall
- IMMUTABLE
- IMS
- INFINITY
- InOne
- ISC
- Jabra
- JBL
- JOCOBOO
- kabeer
- KASODH
- KAWL
- KBOOM
- KDM
- kk2
- KOTION
- KRAZZY
- LIFE


Distribution of Discounts
Frequency
Discount (%)
300
250
200
150
100
50
0
−2000
0
2000
4000
6000
8000
10000
12000


Distribution of Number of Ratings
Frequency
Number of Ratings
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e6



LSTM Model:

```
First few rows of the dataset:
                                            Model   Company       Color  \
0             5PLUS 5PHP28 Wired without Mic Headset     5PLUS         Red
1  A R Wireless compatible with Headset Bluetooth...       A R         Red
2  Aerizo Wireless Touch R100 Earbuds (Black) Blu...    Aerizo       Black
3  Allmusic powerful driven bass with dynamic bea...  Allmusic  Multicolor
4  Allmusic OPP.O Ultra HD Sound Premium Bass Spo...  Allmusic       Black

            Type  Average Rating  Number of Ratings  Selling Price  \
0      On the Ear             3.6                101            496
1      Multicolor             3.9              35280            188
2   True Wireless             4.0               1934            589
3      In the Ear             4.0              15841            260
4      In the Ear             3.8              10766            270

   Maximum Retail Price  Discount
0                  3399      2903
1                   799       611
2                  1298       709
3                  1599      1339
4                   999       729
Columns in the dataset: Index(['Model', 'Company', 'Color', 'Type', 'Average Rating',
       'Number of Ratings', 'Selling Price', 'Maximum Retail Price',
       'Discount'],
      dtype='object')
```

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 1, 64)	18,432
dropout (Dropout)	(None, 1, 64)	0
lstm_1 (LSTM)	(None, 32)	12,416
dropout_1 (Dropout)	(None, 32)	0
dense (Dense)	(None, 32)	1,056
dense_1 (Dense)	(None, 1)	33

Total params: 31,937 (124.75 KB)

Trainable params: 31,937 (124.75 KB)

Non-trainable params: 0 (0.00 B)

```
Epoch 1/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 7s 42ms/step - loss: 1474453.6250 - mae: 849.4838 - val_loss: 1236835.0000 - val_mae: 806.3074
Epoch 2/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1359536.7500 - mae: 858.5170 - val_loss: 1233972.1250 - val_mae: 804.5908
Epoch 3/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1559743.7500 - mae: 883.2617 - val_loss: 1222412.7500 - val_mae: 797.6158
Epoch 4/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1439939.5000 - mae: 838.7285 - val_loss: 1197319.7500 - val_mae: 781.9575
Epoch 5/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1288500.1250 - mae: 802.6878 - val_loss: 1167025.6250 - val_mae: 762.2919
Epoch 6/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 1442587.6250 - mae: 821.0095 - val_loss: 1138183.0000 - val_mae: 742.9955
Epoch 7/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 1153724.2500 - mae: 751.5475 - val_loss: 1110331.5000 - val_mae: 723.9088
```


```
Epoch 6/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 1442587.6250 - mae: 821.0095 - val_loss: 1138183.0000 - val_mae: 742.9955
Epoch 7/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 1153724.2500 - mae: 751.5475 - val_loss: 1110331.5000 - val_mae: 723.9088
```


```
Epoch 8/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 1238190.7500 - mae: 753.0831 - val_loss: 1082267.5000 - val_mae: 704.1917
Epoch 9/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1155615.0000 - mae: 723.4222 - val_loss: 1053990.2500 - val_mae: 683.9403
Epoch 10/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 1323560.5000 - mae: 723.2020 - val_loss: 1025419.4375 - val_mae: 663.2658
Epoch 11/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 979675.8125 - mae: 636.1963 - val_loss: 996801.7500 - val_mae: 642.7718
Epoch 12/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1311541.3750 - mae: 707.5169 - val_loss: 967829.8125 - val_mae: 621.7673
Epoch 13/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1253350.6250 - mae: 700.7159 - val_loss: 939196.3750 - val_mae: 600.5760
Epoch 14/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1175869.6250 - mae: 639.9809 - val_loss: 910742.7500 - val_mae: 580.3579
Epoch 15/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1197143.8750 - mae: 643.9152 - val_loss: 882392.8125 - val_mae: 563.0694
Epoch 16/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 943107.4375 - mae: 593.0568 - val_loss: 854434.3125 - val_mae: 547.5176
Epoch 17/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 888889.6875 - mae: 588.5693 - val_loss: 827556.5625 - val_mae: 532.7982
Epoch 18/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 788072.6875 - mae: 548.6722 - val_loss: 801893.5000 - val_mae: 518.2816
Epoch 19/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 940684.0625 - mae: 549.3251 - val_loss: 777006.2500 - val_mae: 503.8949
Epoch 20/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 1125206.6250 - mae: 589.8337 - val_loss: 753489.2500 - val_mae: 491.0163
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 646006.8125 - mae: 504.4121
Test Mean Absolute Error: 491.02
```

Logistic Regression:

```
                                               Model   Company       Color  \
0            5PLUS 5PHP28 Wired without Mic Headset     5PLUS         Red
1  A R Wireless compatible with Headset Bluetooth...       A R         Red
2  Aerizo Wireless Touch R100 Earbuds (Black) Blu...    Aerizo       Black
3  Allmusic powerful driven bass with dynamic bea...  Allmusic  Multicolor
4  Allmusic OPP.O Ultra HD Sound Premium Bass Spo...  Allmusic       Black

            Type  Average Rating  Number of Ratings  Selling Price  \
0      On the Ear             3.6                101            496
1      Multicolor             3.9              35280            188
2   True Wireless             4.0               1934            589
3      In the Ear             4.0              15841            260
4      In the Ear             3.8              10766            270

   Maximum Retail Price  Discount
0                  3399      2903
1                   799       611
2                  1298       709
3                  1599      1339
4                   999       729
Test Accuracy: 64.50%
              precision    recall  f1-score   support

    negative       0.62      0.96      0.75       112
     neutral       0.00      0.00      0.00         3
    positive       0.84      0.25      0.38        85

    accuracy                           0.65       200
   macro avg       0.49      0.40      0.38       200
weighted avg       0.70      0.65      0.58       200
```

Week 6-8 Report

(5 Oct- 11 Oct), (12 Oct – 18 OCT), (19 Oct – 25 Oct)

Sentiment Analysis Based on Product Reviews and Customer Segmentation

Abstract

This report outlines the methodologies and findings from the customer segmentation analysis based on the Flipkart headphones dataset. The analysis aims to identify distinct customer groups through K-Means clustering based on key product features, including average rating, number of ratings, selling price, maximum retail price, and discount. The findings will assist in targeted marketing strategies and improved product offerings.

Introduction

In the competitive landscape of e-commerce, understanding customer preferences and behaviors is essential for businesses to enhance their product offerings and marketing strategies. This project focuses on analyzing product reviews and performing customer segmentation on a dataset of headphones from Flipkart. By utilizing clustering algorithms, this study aims to uncover patterns that can inform decision-making processes regarding product development and marketing.

Literature Review

Prior research has highlighted the importance of sentiment analysis and customer segmentation in understanding consumer behavior. Techniques such as K-Means clustering and various dimensionality reduction methods have been employed to classify customers into distinct segments. Studies suggest that clustering algorithms can effectively group customers based on their preferences and purchasing behaviors, allowing companies to tailor their strategies accordingly.

Methodology

The methodology employed in this study can be summarized as follows:

1. **Data Loading and Preprocessing**:
   - The Flipkart headphones dataset was loaded using Pandas, and initial exploration was conducted to understand its structure and data types.
   - Non-numeric columns were identified, and relevant numeric features were converted to the appropriate data types. Rows with missing values were dropped to ensure data quality.
2. **Feature Selection**:
   - The features selected for clustering included average rating, number of ratings, selling price, maximum retail price, and discount.
3. **Feature Scaling**:

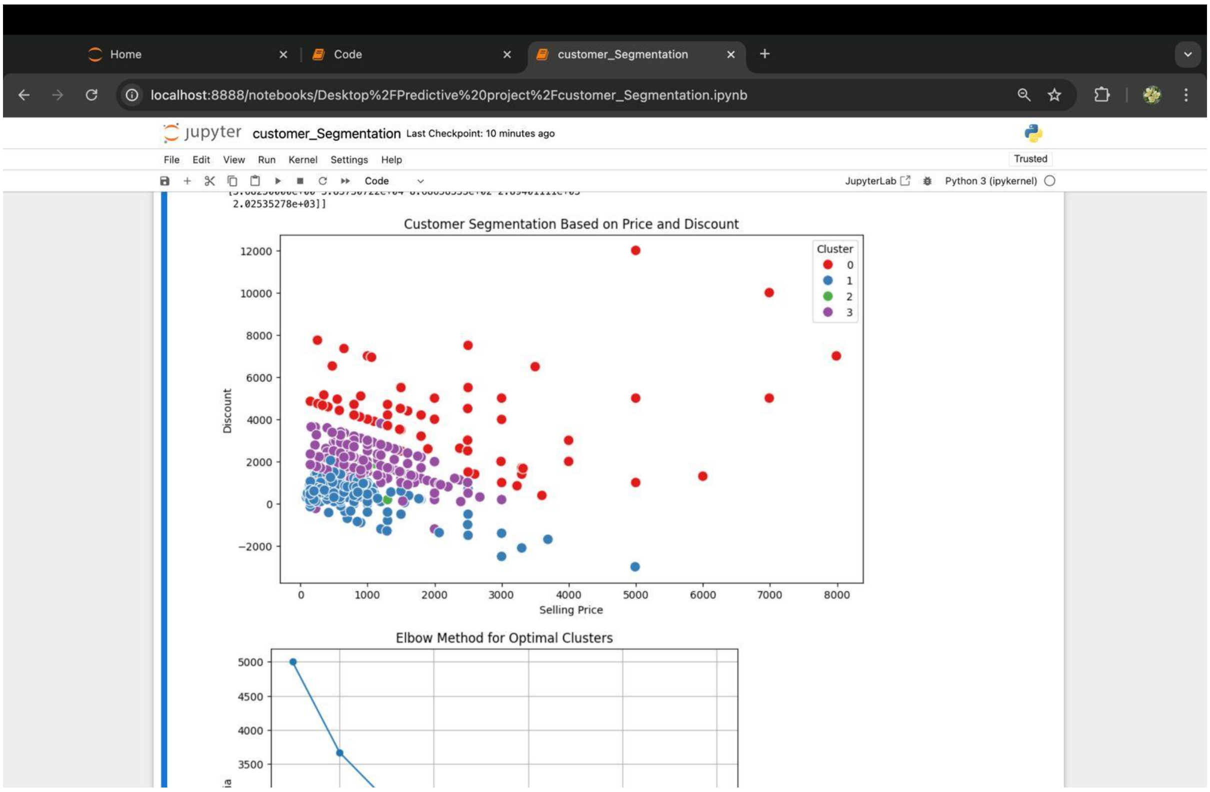
- StandardScaler from scikit-learn was used to scale the features to ensure that each feature contributed equally to the distance calculations in clustering.

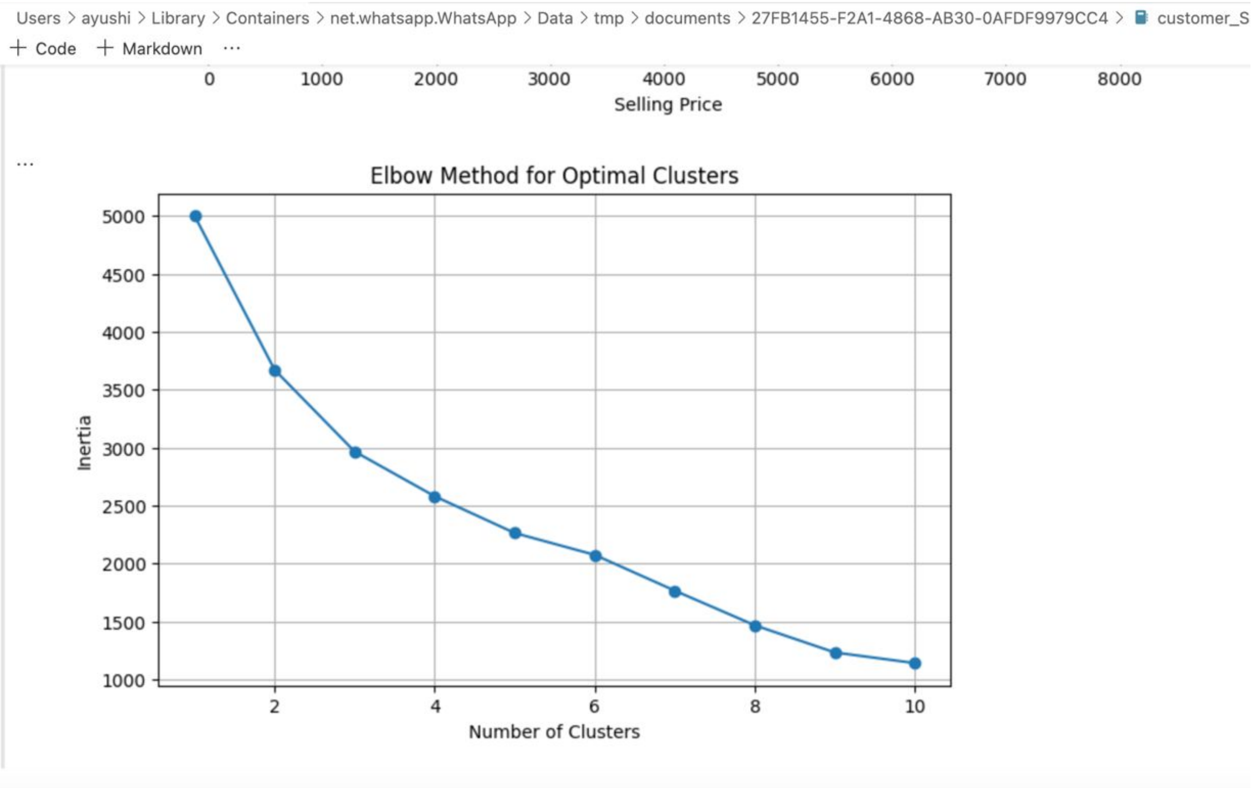
4. **K-Means Clustering**:
   - K-Means clustering was implemented with a pre-defined number of clusters (n_clusters=4). The model was fitted to the scaled data, and cluster labels were assigned to each record.
5. **Visualization**:
   - A scatter plot was generated to visualize the clusters based on selling price and discount, highlighting the segmentation of customers.
6. **Elbow Method**:
   - The Elbow method was employed to determine the optimal number of clusters by plotting the inertia against the number of clusters.

Results

```
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.grid()
plt.show()
```

```
                                            Model   Company       Color  \
0           SPLUS SPHP28 Wired without Mic Headset     SPLUS         Red
1  A R Wireless compatible with Headset Bluetooth...       A R         Red
2  Aerizo Wireless Touch R100 Earbuds (Black) Blu...    Aerizo       Black
3  Allmusic powerful driven bass with dynamic bea...  Allmusic  Multicolor
4  Allmusic OPP.O Ultra HD Sound Premium Bass Spo...  Allmusic       Black

            Type  Average Rating  Number of Ratings  Selling Price  \
0     On the Ear             3.6                101            496
1     Multicolor             3.9              35280            188
2  True Wireless             4.0               1934            589
3     In the Ear             4.0              15841            260
4     In the Ear             3.8              10766            270

   Maximum Retail Price  Discount                        reviews
0                  3399      2903                Meh, it's okay.
1                   799       611  Decent product, but some flaws.
2                  1298       709        Happy with the purchase.
3                  1599      1339             Very good product.
4                   999       729  Reasonable for the price range.
Model                    object
Company                  object
Color                    object
Type                     object
Average Rating          float64
Number of Ratings         int64
Selling Price             int64
Maximum Retail Price      int64
Discount                  int64
reviews                  object
dtype: object
Non-numeric columns: Index(['Model', 'Company', 'Color', 'Type', 'reviews'], dtype='object')
                                            Model   Company       Color  \
0           SPLUS SPHP28 Wired without Mic Headset     SPLUS         Red
1  A R Wireless compatible with Headset Bluetooth...       A R         Red
2  Aerizo Wireless Touch R100 Earbuds (Black) Blu...    Aerizo       Black
3  Allmusic powerful driven bass with dynamic bea...  Allmusic  Multicolor
4  Allmusic OPP.O Ultra HD Sound Premium Bass Spo...  Allmusic       Black

            Type  Average Rating  Number of Ratings  Selling Price  \
0     On the Ear             3.6                101            496
1     Multicolor             3.9              35280            188
2  True Wireless             4.0               1934            589
3     In the Ear             4.0              15841            260
4     In the Ear             3.8              10766            270

   Maximum Retail Price  Discount                        reviews  Cluster
0                  3399      2903                Meh, it's okay.        3
1                   799       611  Decent product, but some flaws.        1
2                  1298       709        Happy with the purchase.        1
3                  1599      1339             Very good product.        1
4                   999       729  Reasonable for the price range.        1
Cluster Centers (scaled):
 [[-0.00749104 -0.21192749  1.50481062  1.9858429   1.71482007]
 [ 0.19452639 -0.14031077 -0.38218698 -0.65345361 -0.63270913]
 [ 0.83841287  5.54701154 -0.27773529 -0.38644752 -0.34285457]
 [-0.31783417 -0.07424497  0.04406116  0.26561277  0.32459316]]
Cluster Centers (original scale):
 [[3.82750000e+00 1.67361167e+04 2.05497500e+03 5.94421667e+03
  3.88924167e+03]
 [3.92188755e+00 2.79907610e+04 5.22489960e+02 1.26437952e+03
  7.41889558e+02]
 [4.22272727e+00 9.21759545e+05 6.07318182e+02 1.73781818e+03
  1.13050000e+03]
 [3.68250000e+00 3.83730722e+04 8.68658333e+02 2.89401111e+03
  2.02535278e+03]]
```


Home
Code
customer_Segmentation
localhost:8888/notebooks/Desktop%2FPredictive%20project%2Fcustomer_Segmentation.ipynb
jupyter customer_Segmentation Last Checkpoint: 10 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
2.02535278e+03]]
Customer Segmentation Based on Price and Discount
Cluster
0
1
2
3
Discount
Selling Price
Elbow Method for Optimal Clusters



Conclusion

The clustering analysis revealed significant patterns in customer preferences regarding headphones on Flipkart. By identifying distinct segments, businesses can tailor their marketing strategies and product offerings to meet the needs of different customer groups. Future work could explore integrating sentiment analysis from product reviews to enhance the segmentation process further.

References

- Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques. Elsevier.
- Jain, A. K. (2010). Data Clustering: 50 Years Beyond K-Means. *Pattern Recognition Letters*, 31(8), 651-666.
- Xu, R., & Wunsch, D. (2010). Clustering. *Wiley Encyclopedia of Computer Science and Engineering*.

Code

```
# Import necessary libraries

import pandas as pd

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans
```

```
import matplotlib.pyplot as plt

import seaborn as sns


# Load the dataset

# Assuming the dataset is a CSV file named 'flipkart_headphones.csv'

data = pd.read_csv('Flipkart Headphones.csv')


# Display the first few rows of the dataset

print(data.head())


# Check data types of the columns

print(data.dtypes)


# Identify non-numeric columns

non_numeric_columns = data.select_dtypes(include=['object']).columns

print("Non-numeric columns:", non_numeric_columns)


# Convert numeric columns to the appropriate data type

data['Average Rating'] = pd.to_numeric(data['Average Rating'], errors='coerce')

data['Number of Ratings'] = pd.to_numeric(data['Number of Ratings'], errors='coerce')

data['Selling Price'] = pd.to_numeric(data['Selling Price'], errors='coerce')

data['Maximum Retail Price'] = pd.to_numeric(data['Maximum Retail Price'], errors='coerce')

data['Discount'] = pd.to_numeric(data['Discount'], errors='coerce')


# Drop rows with any NaN values that may have resulted from conversion

data = data.dropna(subset=['Average Rating', 'Number of Ratings', 'Selling Price', 'Maximum Retail
Price', 'Discount'])


# Select relevant features for clustering

features = ['Average Rating', 'Number of Ratings', 'Selling Price', 'Maximum Retail Price', 'Discount']
```

```
# Feature scaling
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data[features])

# K-Means clustering implementation
kmeans = KMeans(n_clusters=4, random_state=42)  # You can adjust the number of clusters
kmeans.fit(data_scaled)

# Add the cluster labels to the original dataset
data['Cluster'] = kmeans.labels_

# Display the clustered data
print(data.head())

# Check the cluster centers (in the scaled space)
print("Cluster Centers (scaled): \n", kmeans.cluster_centers_)

# Inverse transform the cluster centers to the original scale
print("Cluster Centers (original scale): \n", scaler.inverse_transform(kmeans.cluster_centers_))

# Plot the clusters
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Selling Price', y='Discount', hue='Cluster', data=data, palette='Set1', s=100)
plt.title('Customer Segmentation Based on Price and Discount')
plt.xlabel('Selling Price')
plt.ylabel('Discount')
plt.legend(title='Cluster')
plt.show()

# Elbow method to find the optimal number of clusters
inertia = []
```

```
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(data_scaled)
    inertia.append(kmeans.inertia_)

# Plot the Elbow curve
plt.figure(figsize=(8, 5))
plt.plot(range(1, 11), inertia, marker='o')
plt.title('Elbow Method for Optimal Clusters')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.grid()
plt.show()
```