InkSight Analyzer

Fresh Ink on Insights

EBA5004 Graduate Certificate in Practical Language Processing
Sun, 27 Oct 2024

Ang **Mei Chi**	(A0269733R)
Darius Chan Yak Weng	(A0243799J)
Lew Kuan Teng **Roy (PM)**	(A0124354M
Liu **Wudi**	(A0269809J)
Michael Wong Wait Kit	(A0269491N)

Presentation Outline

Wudi

1. Introduction

   Background, Business Objective, Data Source & EDA

MC

2. Data Architecture, Preprocessing & Initial Models

   Model Exploration (Aspect)

Roy
(PM)

3. Model Analysis & Evaluation

   Model Exploration (Sentiment) and Final Model

Mich

4. System Demo

   Streamlit Interface

DC

5. Conclusion

   Challenges, Limitations & Retrospective

Introduction

Background, Business Objective, Data Source & EDA

Background

HP Printing Business

- **Top 3** home/office printer brand in the world *(alongside Epson and Xerox)*

- Highest market share @**35%**

- Values customer input as top priority to drive continuous improvement and innovation

Data Sources

Origins of Customer Comments

Web Reviews – Star Ratings (1-5 stars)

Insights:

- Customer reviews
- Survey verbatim

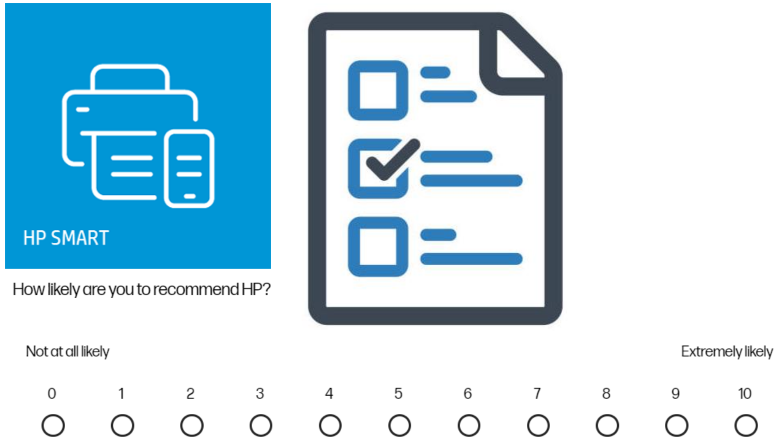
Key metrics:

- Star Ratings (1-5 stars)
- LTR Score (0-10)

Existing Sentiment Grouping

Positive: ≥4 stars / ≥6 LTR score

Negative: <4 star / <6 LTR score



HP SMART App Surveys – **L**ikelihood **T**o **R**ecommend (**LTR**) Score (0-10)

Project Objective

Customers First Sentiment and Aspects

Actionable insights → Conduct an in-depth analysis of customer sentiment

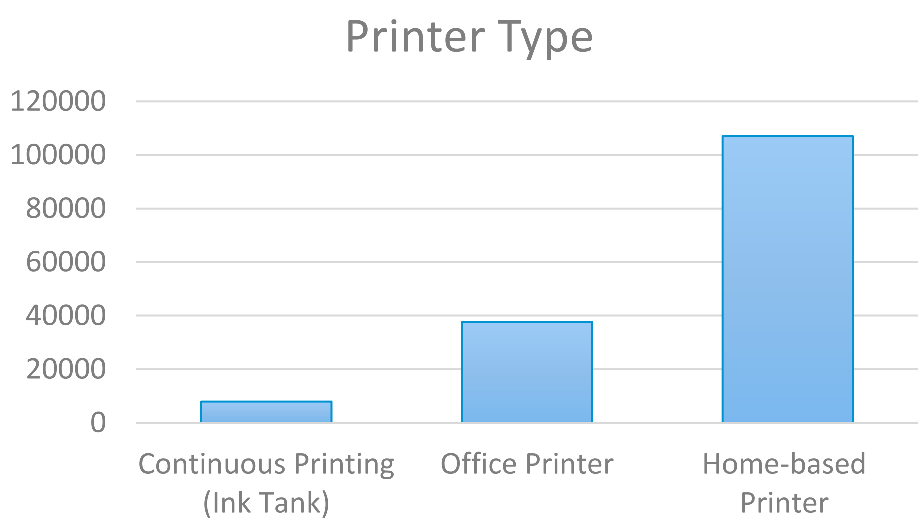
Identify Sentiments & Aspects

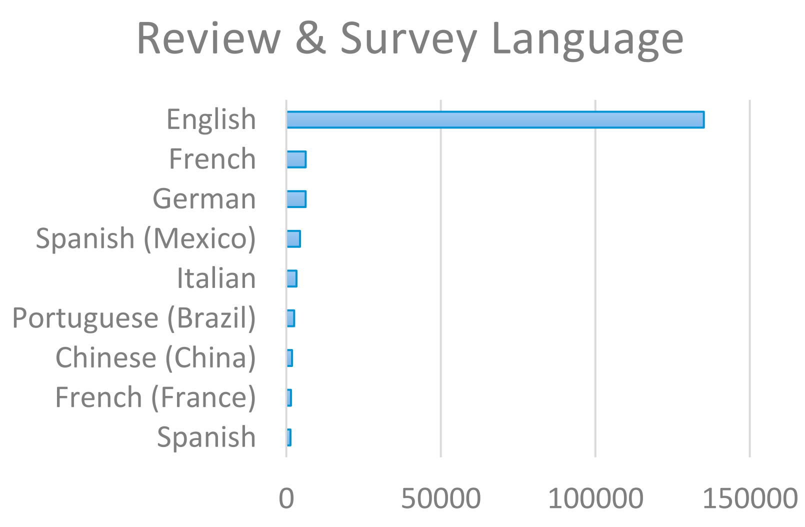
Uncovering **pain points** for insights generation

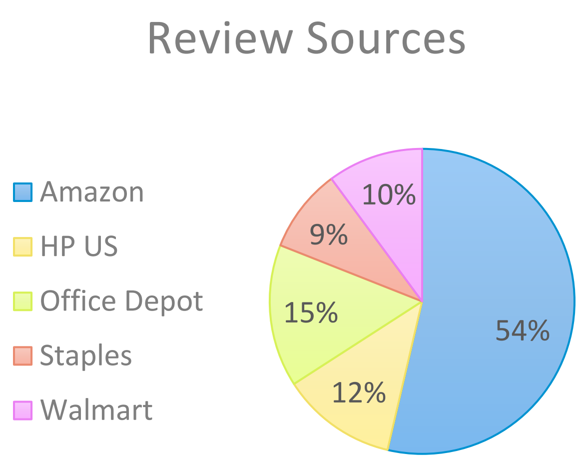
Get actionable insights and inform product development, marketing strategies and customer support.

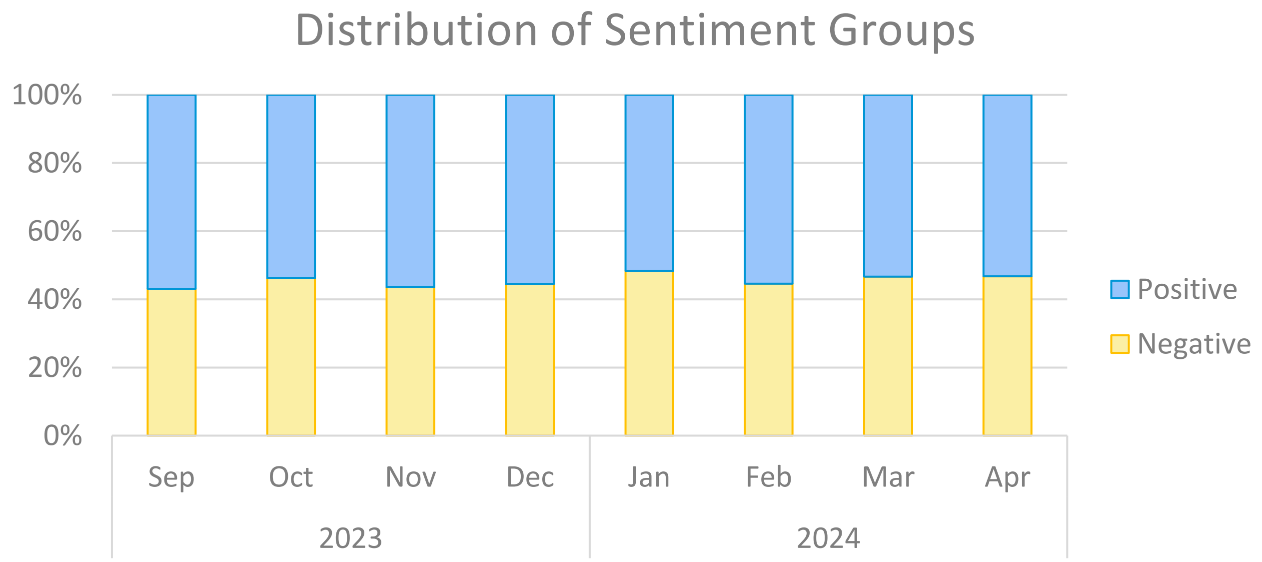
Exploratory data analysis

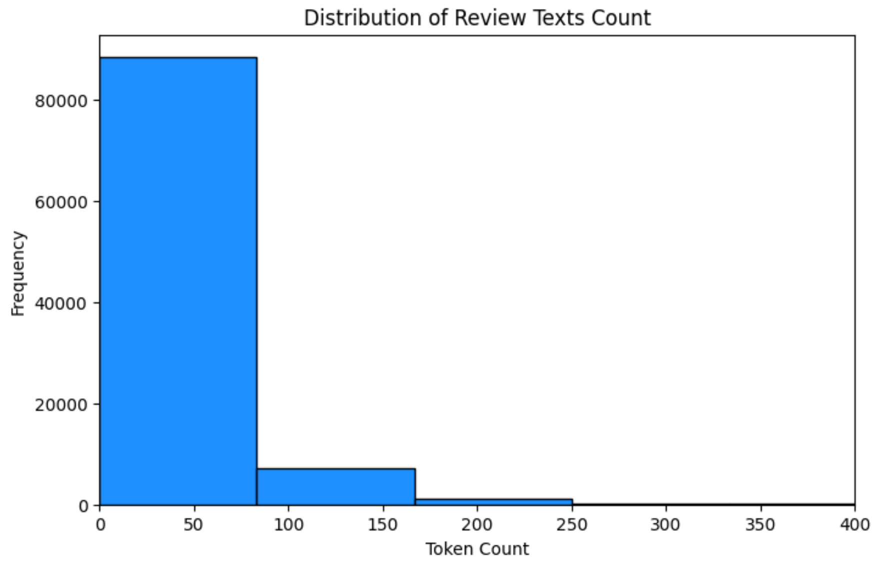
Data Distribution





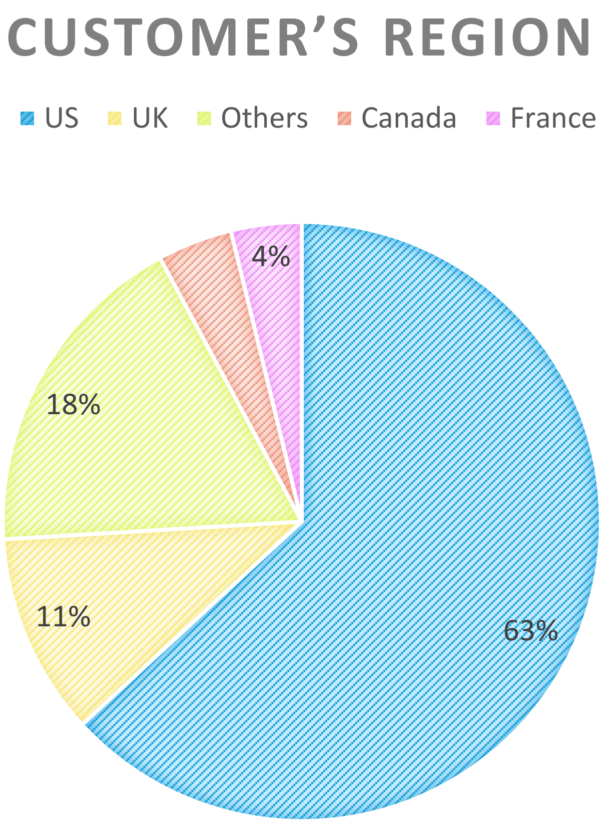


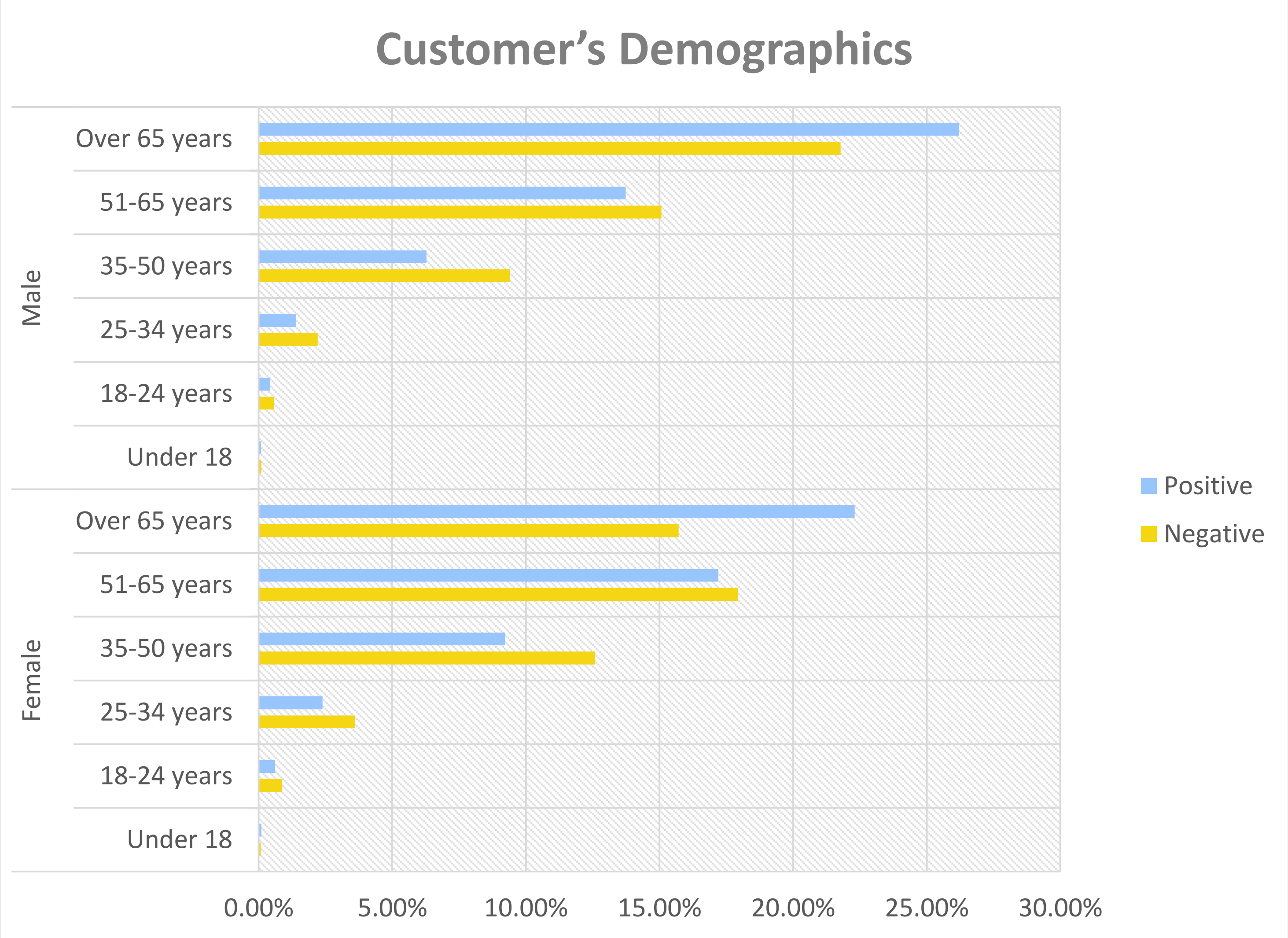




Survey Demographics

Who're the customers?





Binning and Recategorization

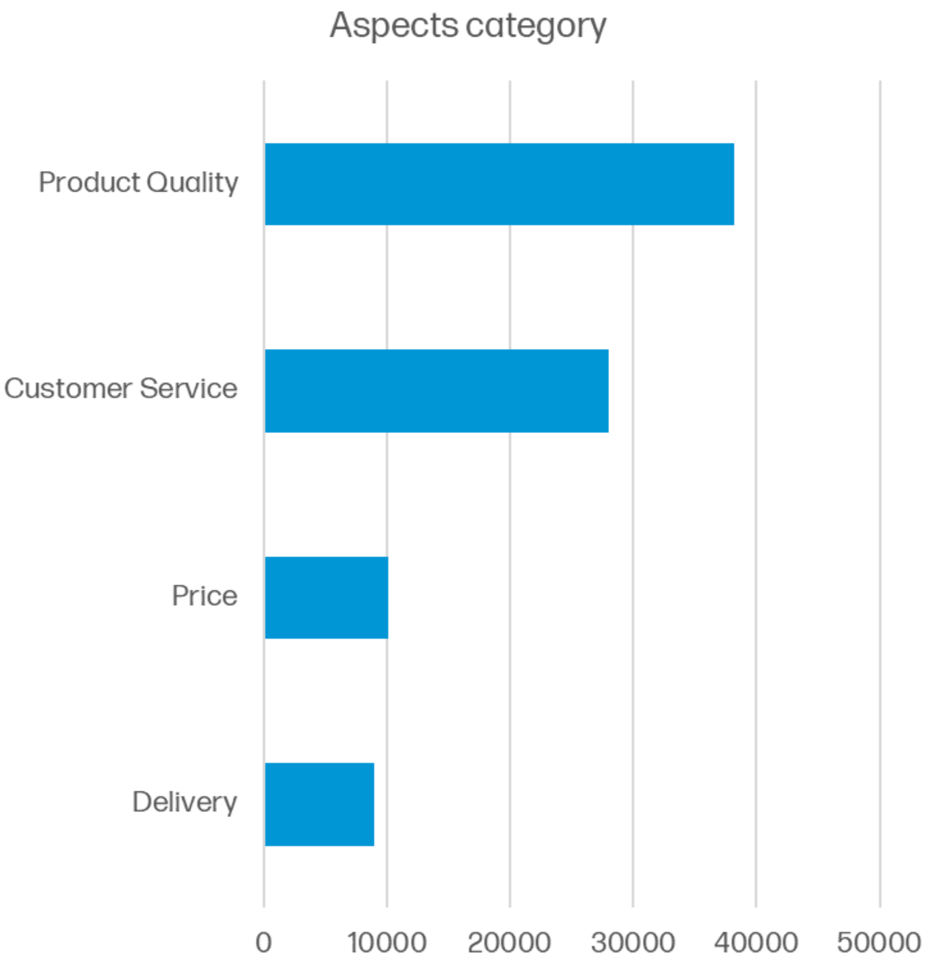
Key pain points -> Valuable Aspects Labels

- Printer & Ink Quality
- Cartridge problem
- Print output quality

- Customer experience

- Cost

- Logistics handling
- Cartridge Shipment

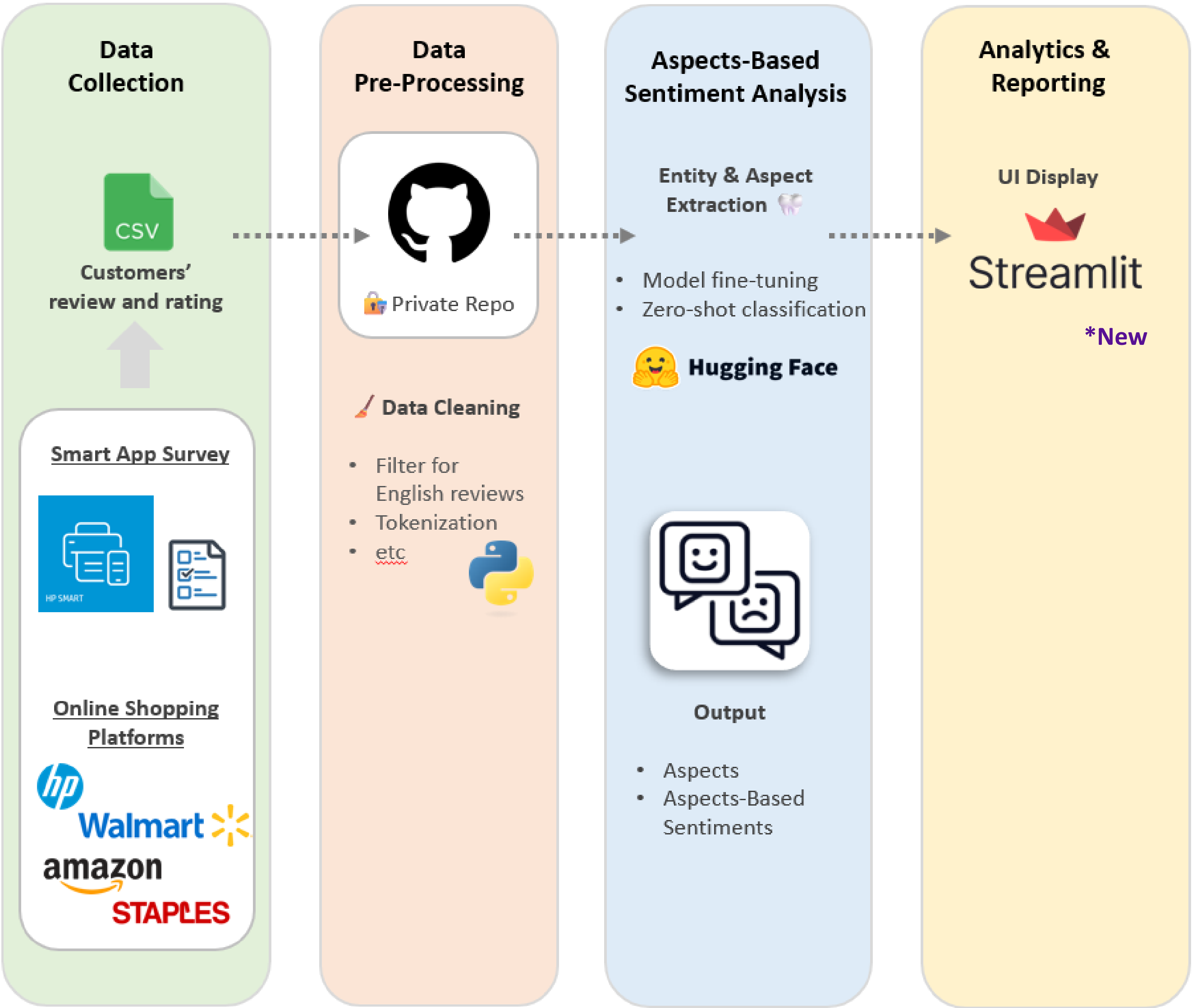


Data Architecture, Preprocessing & Initial Models

Model Exploration (Aspect)

Data Architecture

Overview



- *Customizable*
- *Developer-friendly*
- *Fast & Interactive*

Data Pre-Processing

Key steps

Reduce number of aspect categories

- From 10 to 4 key aspects

Original 10 Aspects

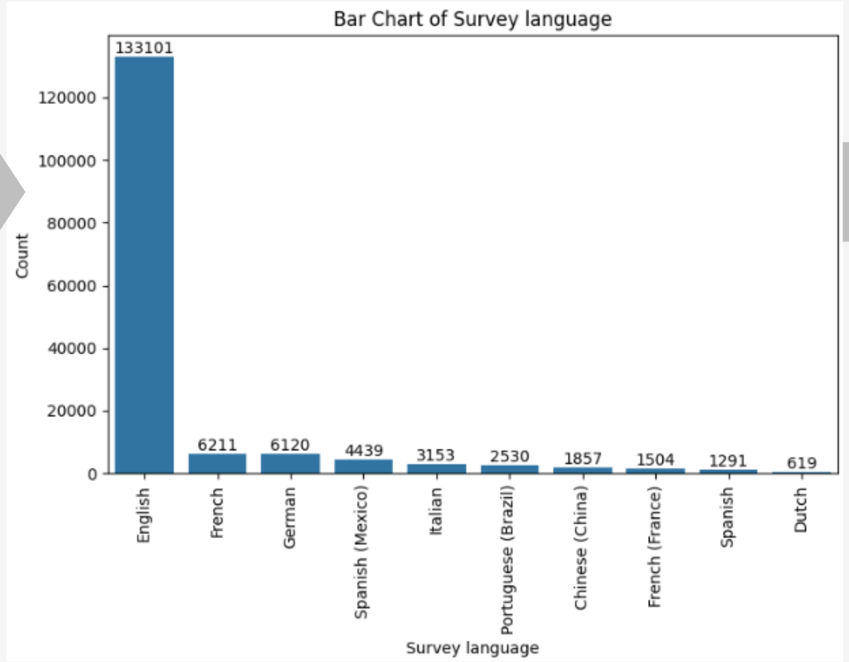
- Brand loyalty
- Cartridge problem
- Cost
- Customer experience & expectation
- General satisfaction
- Instant ink program
- Logistic handling
- Print quality
- Printer hardware / printing issues
- Others

4 Key Aspects

- Price
- Customer service
- Product quality
- Delivery

Filter 'Survey Language' for 'English'

- Models predominant in English



Use for model training, testing & validation

- Original dataset : Jun'23 – Apr'24 Data
- Training & testing dataset : prior to Mar'24
- Validation dataset : Apr'24

To ensure data **recency** and **relevance**

End-to-End vs Modular ABSA

End-to-End System

Clean Dataset

↓

Perform sentiment analysis for each of the aspects

Modular System

Clean Dataset

↓

Identify the **aspects**

↓

Extract sentiment expressions for each of the aspects identified

↓

Perform sentiment analysis for each of the aspects

End-to-End

Natural Language Inference

- Multi-Turn Zero-Shot ABSA

Multi-Turn Zero-Shot ABSA

Strategies

First Inference

Loop through for 4 key aspect categories

Aspect : Price

- This is a comment about Price
- This is not a comment about Price

Second Inference

Loop through for the aspects identified from the first inference

Aspect Identified : Price

- This is a positive comment about Price
- This is a negative comment about Price

Multi-turn Zero-shot ABSA

Multi-Label Natural Language Inference

fbBART

facebook/bart-large-mnli (~400m param):

- Based on BART, which excels in sequence-to-sequence tasks and text generation.
- Fine-tuned on the **MNLI** dataset, making it highly effective in handling entailment and contradiction.
- Strong at both understanding and generating complex text due to its pre-training.

DeBERTa

MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli (~180m param):

- DeBERTa's enhanced attention and positional encoding improve contextual understanding.
- Fine-tuned on **MNLI**, **FEVER**, and **ANLI**, providing robustness across different inference tasks.
- Excellent at capturing long-range dependencies, improving overall performance.

Datasets
MNLI - Multi-Genre Natural Language Inference
FEVER - Fact Extraction and VERification
ANLI - Adversarial Natural Language Inference
SNLI - Stanford Natural Language Inference corpus

roBERTa

cross-encoder/nli-roberta-base (~60m param):

- Uses RoBERTa, an optimized version of BERT, with better training techniques.
- Cross-encoder architecture processes sentence pairs together for more accurate classification.
- Fine-tuned on **SNLI** with specifically for NLI tasks, making it well-suited for binary entailment and contradiction.

Performance Evaluation

Aspects Identification

Data	Metrics	End-to-End System (Natural Language Inference)								
		Facebook BART large mnli			DeBERTa-v3-base-mnli-fever-anli			cross-encoder/nli-roberta-base		
		micro avg	macro avg	weighted avg	micro avg	macro avg	weighted avg	micro avg	macro avg	weighted avg
Mar'24	Accuracy	68.6%	68.6%	63.6%	70.7%	70.7%	65.2%	72.2%	72.2%	67.9%
	Precision	48.3%	54.2%	51.6%	51.8%	57.9%	54.0%	61.9%	69.1%	66.9%
	Recall	50.9%	51.9%	50.9%	46.6%	49.3%	46.6%	21.7%	23.8%	21.7%
Apr'24	Accuracy	61.5%	61.5%	57.8%	62.7%	62.7%	58.1%	64.2%	64.2%	60.3%
	Precision	48.7%	48.6%	47.6%	50.8%	52.5%	49.2%	60.1%	60.1%	57.4%
	Recall	39.3%	46.9%	39.3%	35.5%	42.7%	35.5%	15.4%	19.1%	15.4%

- NLI model performed poorly in identifying the aspects targeted in the reviews
- Very low recall rate suggests a high likelihood of missed insights for HP Management.

Legends:

< 50%

> 70%

Performance Evaluation

Aspects-Based Sentiment Analysis

Data	Metrics	Natural Language Inference (E2E)					
		Facebook BART large mnli		DeBERTa-v3-base-mnli-fever-anli		cross-encoder/nli-roberta-base	
		Positive	Negative	Positive	Negative	Positive	Negative
Mar'24	Accuracy Rate	35.3%	35.3%	33.6%	33.6%	15.6%	15.6%
	Precision Rate	40.6%	32.3%	36.8%	30.5%	17.8%	12.8%
	Recall Rate	29.8%	41.0%	31.7%	36.3%	14.9%	17.2%
Apr'24	Accuracy Rate	34.0%	34.0%	31.6%	31.6%	14.3%	14.3%
	Precision Rate	38.1%	31.6%	34.6%	29.0%	16.0%	12.0%
	Recall Rate	29.2%	38.6%	29.5%	34.4%	13.8%	15.5%

- The downstream butterfly effect from inaccurate aspect identification severely impacted the overall ABSA performance, resulting in very poor sentiment analysis outcomes.

Legends:

< 50%

> 70%

Modular System

Aspects Identification

- Lexico-Syntactic Approach

- Model Fine-Tuning Approach:
  - **Multiclass Classifier**
  - **Multilabel Classifier**

Lexico-Syntactic Approach

Strategies

Pre-processing using spaCy library

- Tokenize the review
- Part-of-Speech (POS) tagging
- Identify syntactic relationships between the word tokens

Extracting aspects and sentiment expressions using Dependency Pairs

- Amod (adjectical modifier)
- Advmod (adverbial modifier)
- Det (determiner)

Map nouns, noun phrases or verbs to aspects; and map their modifiers to sentiment expressions

- Verb : Shipped > Delivery
- Modifier : Fast > Positive Sentiment

Lexico-Syntactic Approach

Findings

Observation

- **Aspect Disambiguation** :
  - After extracting noun phrases (aspects), these need to be mapped to a predefined list of aspect categories to ensure consistency.
  - For example, the terms "price" and "cost" should map to the same aspect, "Price“

- **Irrelevant Dependency Pairs** :
  - Further filtering mechanisms are required to remove irrelevant pairs (e.g., ‘a’ ‘cartridge’).
  - This process demands substantial manual effort to prepare handcrafted rules or filtering mechanisms, which need constant updates as new dependency patterns and review structures emerge. Crafting these rules is time-consuming and prone to error.

- **Implicit Aspect and Sentiment Expression Extraction** :
  - This approach misses implicit aspects or sentiment expressions, which are not directly named.
  - For instance, in reviews like "Only problem was it took more than a week to receive them," the aspect "Delivery" is implied but not explicitly mentioned, hence this approach fails to identify the "Delivery" aspect.

Model Fine-Tuning Approach

Overview

Mar’24 Reviews

- 80% Training
- 20% Testing

Model Fine-Tuning:

DistilBERT

- Model = DistilBertForSequenceClassification ('distilbert-base-uncased’)
- Optimizer : adam
- Loss function : Binary Cross-Entropy Loss (BCEWithLogitsLoss)
- Batch size : 16
- Learning rate : 5e-5
- Epochs : 3

Evaluation:

- Using Apr’24 reviews

Multiclass Classifier
Overview

Initial Observation :

- Only have one LTR or Star Rating per one review
- Does not have the sentiment score for different aspects mentioned in the reviews
- Initial Model : Focus on single aspect sentiment analysis based on 80-20 pareto principle

Existing Dataset :

- More than 75% of the reviews are focusing on single aspect

To include only

Reviews with Single Aspect :

- Shorter character length (~ <20)
- 1 Sentence

Used for

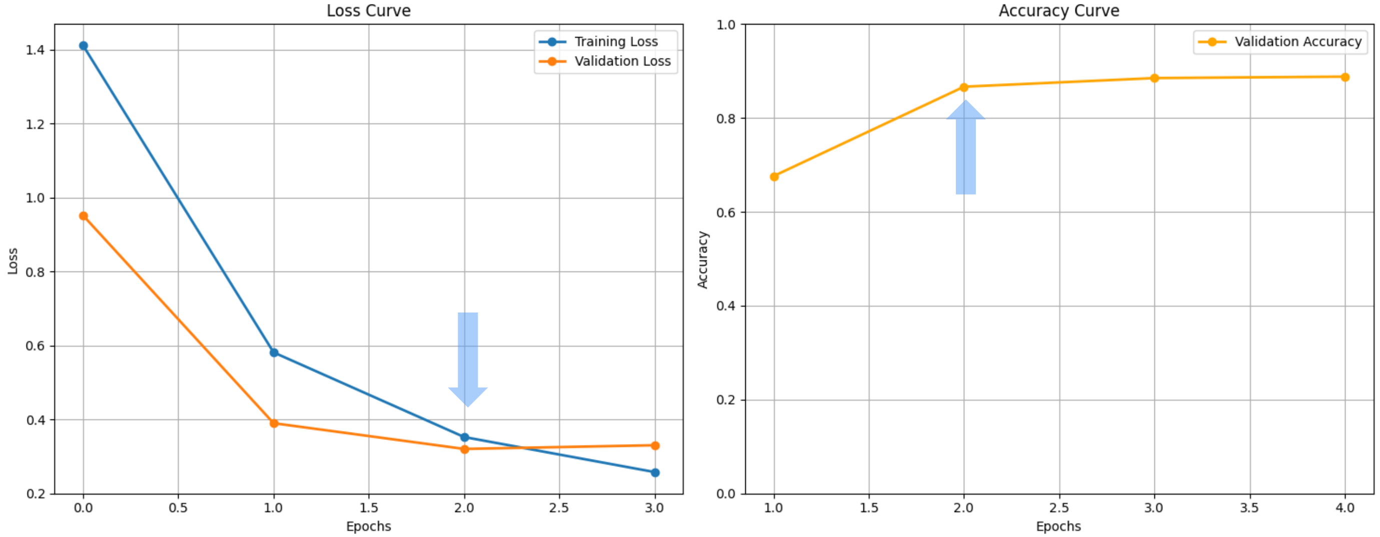
Model Fine-Tuning

DistilBERT

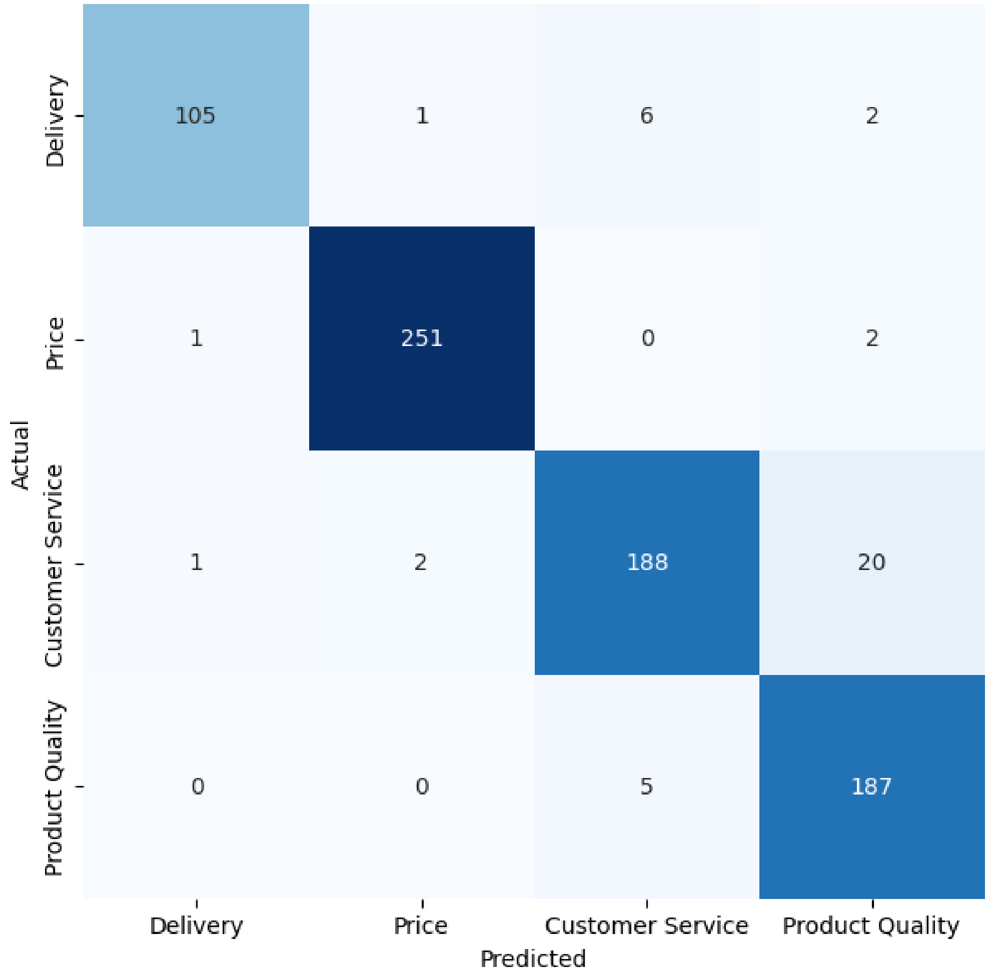
Multiclass Classifier

Single Aspect Classification

Loss and Accuracy Curve



Confusion Matrix



Hyperparams

- Epoch = 3
- Batch Size = 16
- Loss = Cross Entropy
- Evaluation = Steps

Model Summary

	Category	Accuracy	Precision	Recall
0	Delivery	0.921053	0.981308	0.921053
1	Price	0.988189	0.988189	0.988189
2	Customer Service	0.890995	0.944724	0.890995
3	Product Quality	0.973958	0.886256	0.973958

Overall Accuracy for model_1_predictions: $\boxed{0.9481}$

Model Accuracy of ~94%

Multiclass Classifier

Single Aspect Classification

Review	Aspect Category	Model output score
The ink dries as soon as it is replaced, surprisingly fast	**Customer Service**	Customer Service 0.97, Delivery 0.01, Price 0.01, Product Quality 0.00
Lasts for several print jobs	**Customer Service**	Customer Service 0.93, Delivery 0.00, Price 0.00, Product Quality 0.06
happy with the printer,but probems signing on to receive ink	Delivery	Customer Service 0.01, Delivery 0.98, Price 0.00, Product Quality 0.00
they charge monthly and never received the ink cartages	Delivery	Customer Service 0.01, Delivery 0.98, Price 0.00, Product Quality 0.00
Ink cartridges don’t last long enough and are way to expensive	Price	Customer Service 0.01, Delivery 0.00, Price 0.99, Product Quality 0.00
Reasonable price, ease to set up, good performance with instant ink support.	Price	Customer Service 0.01, Delivery 0.00, Price 0.92, Product Quality 0.06
It was easy set up and make copies fast and great color prints also	Product Quality	Customer Service 0.01, Delivery 0.00, Price 0.00, Product Quality 0.98
Ink cartridge or printer appears to be defective - keep getting error	Product Quality	Customer Service 0.02, Delivery 0.00, Price 0.00, Product Quality 0.97

Abstract and Nuanced nature of Customer Service

Multilabel Classifier

Overview

Existing Dataset

NO EXCLUSION →

All Reviews including those with multiple aspects

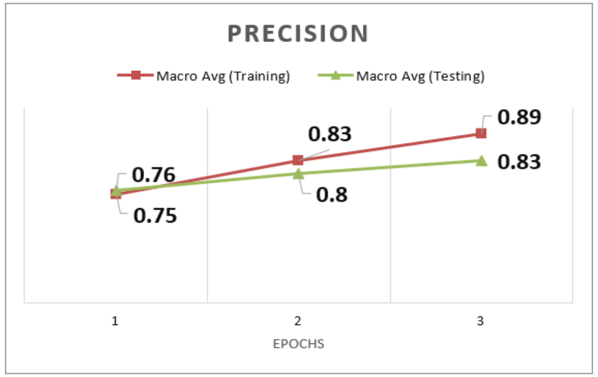
Used for →

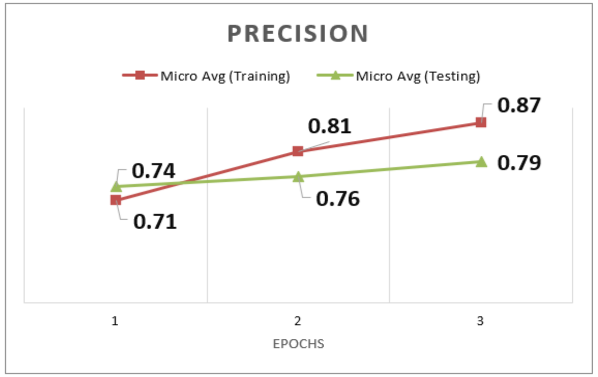
Model Fine-Tuning

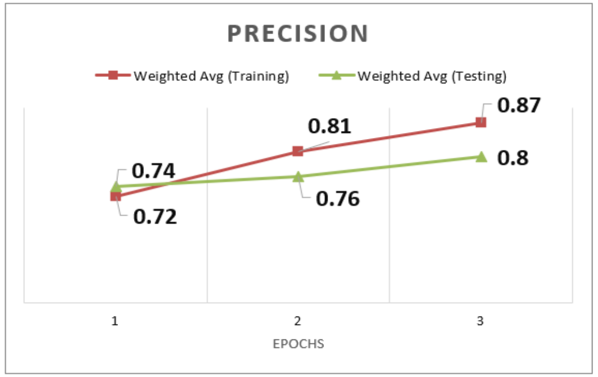
DistilBERT

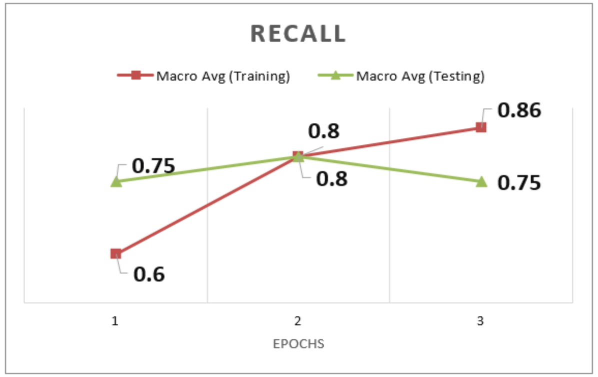
Multilabel Classifier

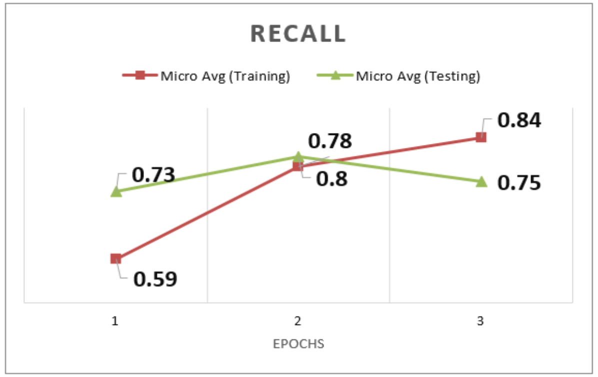
Model Performance

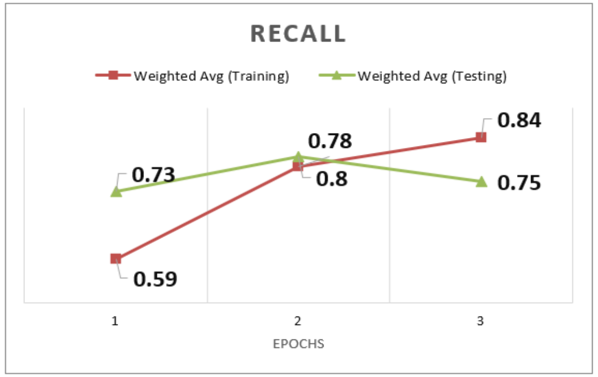


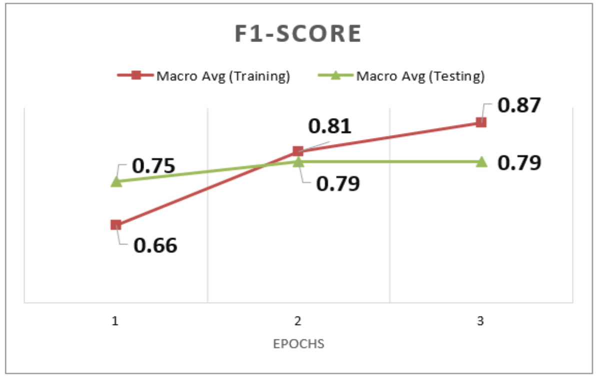


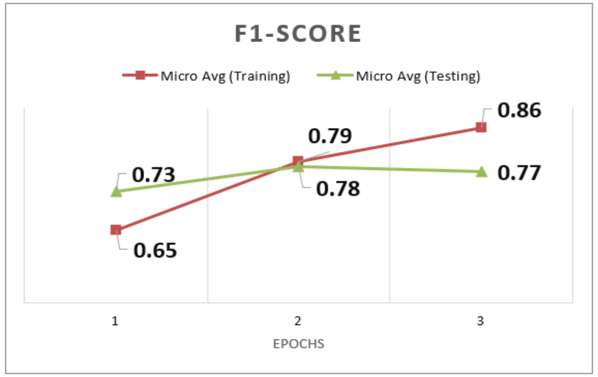


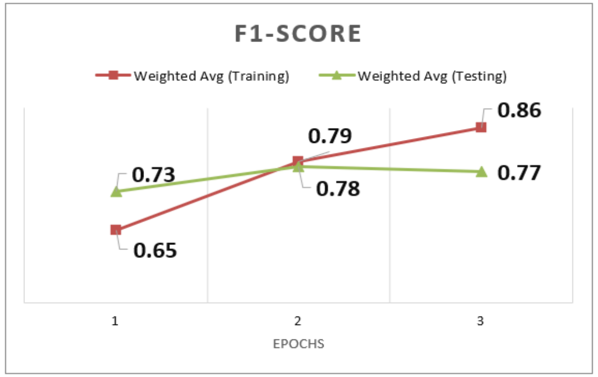












- The model's performance is strong, achieving over 75% across all metrics for both the training and testing datasets

- Hence, we will deploy multilabel classifier to production for multiple aspects identification purpose

Model Analysis and Evaluation

Sentiment Assessment and Final Model

Model Exploration

Sentiment Expression Extraction

Models Explored

- Llama 3.1
- DistilBERT for Question Answering
- Distilbert-base-cased-distilled-squad

Sentiment Expression Extraction

Llama 3.1 Model

	Llama3.1
Technique	Few-Shots
Prompts	Extract the sentiment expression for each aspect mentioned in the review. Return the output in this json format below. Return the json only, no other text

Sample Output :


```
{
  "review": "The new smartphone has a great camera and
  "aspects": [
    {
      "name": "Battery Life",
      "sentiment_expression": "decent"
    },
    {
      "name": "Storage",
      "sentiment_expression": "too small"
    }
  ],
  "Review": "I placed my order for shipping but apparen
  "aspects": [
    {
      "name": "Delivery",
      "sentiment_expression": null
    }
  ]
}
```

```
{
  "review": "The new smartphone has a great camera and decent batter
  "aspects": [
    {
      "aspect": "Battery Life",
      "sentiment_expression": "decent"
    },
    {
      "aspect": "Storage",
      "sentiment_expression": "too small"
    }
  ]
}
```

```
{
  "review": "The quality is good but the price is expensive.",
  "aspects": [
    {
      "aspect": "quality",
      "sentiment_expression": "good"
    },
    {
      "aspect": "price",
      "sentiment_expression": "expensive"
    }
  ]
}
```

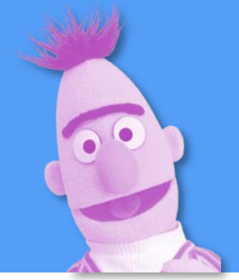

- Inconsistent JSON output format
- Output included examples provided

Observation:

- Not optimized for sentiment extraction
- Inefficient compared to smaller task-specific models
- Fine-tuned models handle sentiment nuances better

Sentiment Expression Extraction

DistilBERT for Question Answering



	DistilBERT - QA
Technique	Zero-Shot
Prompts	What is the sentiment expression for the aspect?

Sample Output :

```
# Example usage
review = "The quality is good but the price is expensive."
aspect = "quality"
```

```
Sentiment expression for 'quality':
Sentiment expression for 'price': [SEP]
```

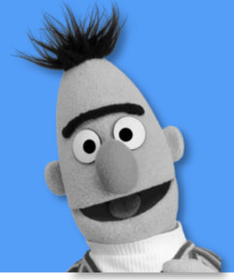
- **Blank / [SEP] output**

Observation:

- DistilBERT is optimized for question-answering, not sentiment analysis or extraction
- It lacks fine-tuning for sentiment-specific tasks, making it less accurate for detecting emotion
- QA models focus on finding factual answers, not understanding sentiment nuances

Sentiment Expression Extraction

distilbert-base-cased-distilled-squad



	distilbert-base-cased-distilled-squad
Technique	Zero-Shot
Prompts	• What is the sentiment expression for the aspect? • Only show sentiment if score > 0.5, else show ‘Not Found’

Sample Output :

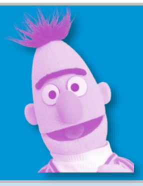
```
Review: 'Fast ship good price. It's a shame I had to
order this from another walmart and not my local store'
Sentiment Expressions: {'Price': 'Not found', 'Ship': 'Not found'}
```

- **Majority ‘Not Found’**
- **Unable to identify sentiments**

Observation:

- Specifically trained for question-answering, not zero-shot sentiment classification
- Lacks the necessary generalization required for sentiment tasks without fine-tuning on sentiment data
- Limited zero-shot capabilities since the model is not pre-trained to handle diverse sentiment labels directly

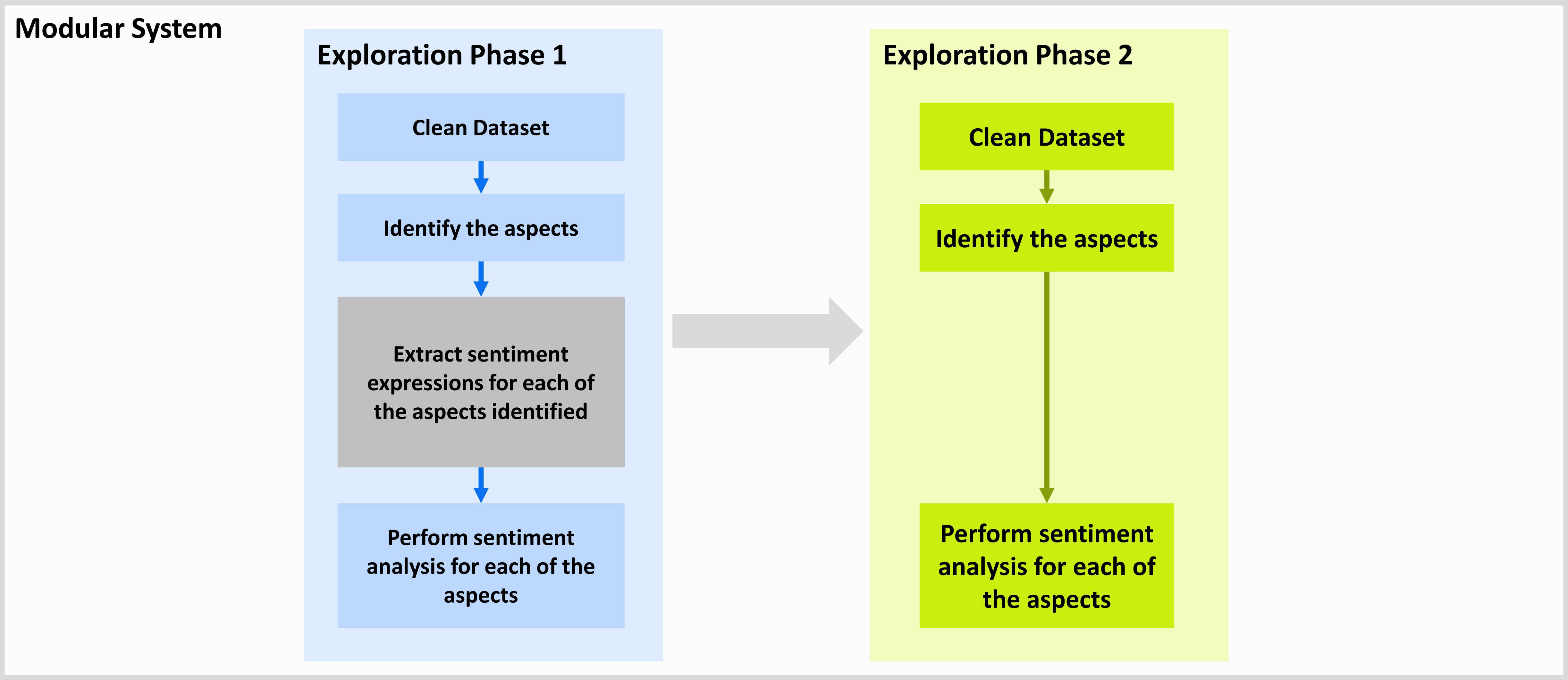
Sentiment Expression Extraction

	Llama3.1	DistilBERT - QA 	distilbert-base-cased-distilled-squad 
Technique	Few-Shots	Zero-Shot	• Zero-Shot
Prompts	Extract the sentiment expression for each aspect mentioned in the review. Return the output in this json format below. Return the json only, no other text	What is the sentiment expression for the aspect?	• What is the sentiment expression for the aspect? • Only show sentiment if score > 0.5, else show ‘Not Found’
Observation	• **Inconsistent JSON output format** • **Output included examples provided**	• **Blank / [SEP] output**	• **A lot of ‘Not Found’**

Observation :

Without labeled training data for fine-tuning a pre-trained model, relying solely on prompt engineering is unlikely to extract sentiment expressions effectively.

Revised Strategy (Modular System)



Sentiments Analysis

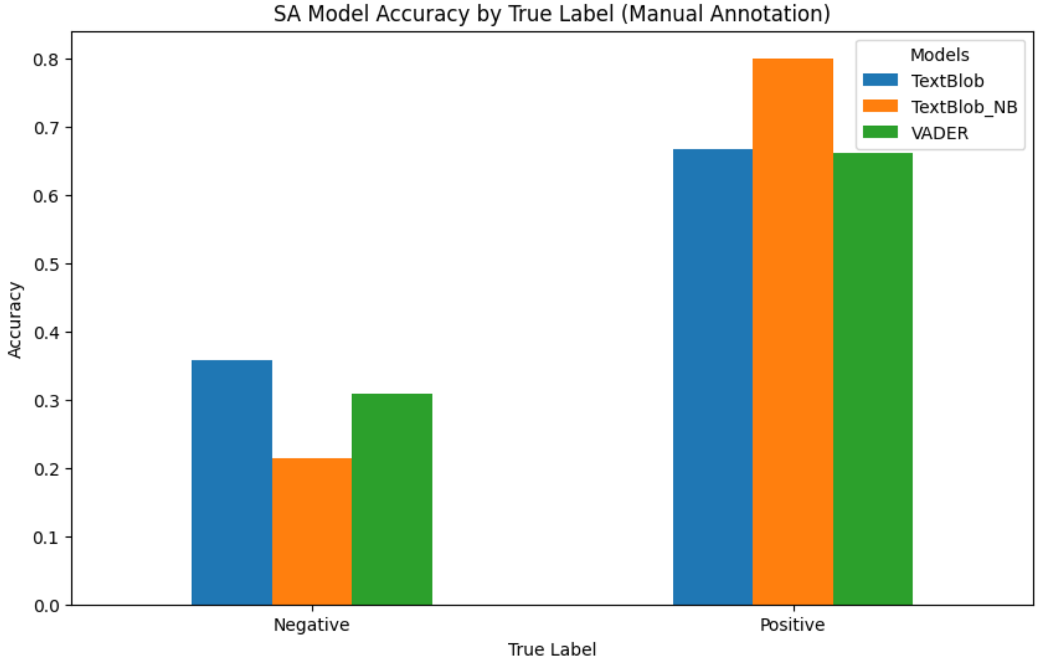
Single Aspect

Lexicon-based approach (~60% accuracy)

- TextBlob
- Vader

Overall Accuracy

```
TextBlob Accuracy: 0.5670859538784067
VADER Accuracy: 0.5482180293501048
TextBlob NB Accuracy: 0.6121593291404612
```



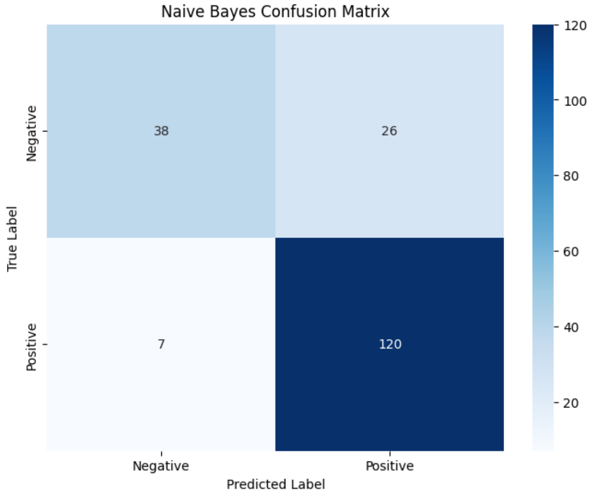
Machine Learning approach (~80% accuracy)

(Trained model using annotated labels @1k sample, at 80:20 split)

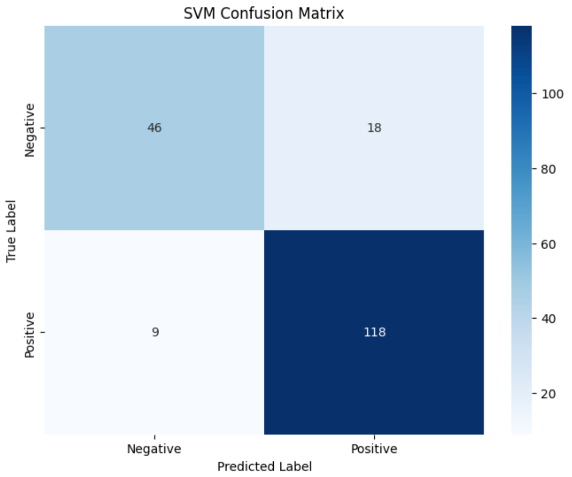
- Naïve Bayer
- SVM

Overall Accuracy

```
Naive Bayes accuracy: 0.8272251308900523
Naive Bayes classification report:
              precision    recall  f1-score   support

    Negative       0.84      0.59      0.70        64
    Positive       0.82      0.94      0.88       127

    accuracy                           0.83       191
   macro avg       0.83      0.77      0.79       191
weighted avg       0.83      0.83      0.82       191
```




```
SVM Model Accuracy: 0.858639
              precision    recall  f1-score   support

    Negative       0.84      0.72      0.77        64
    Positive       0.87      0.93      0.90       127
     Neutral       0.00      0.00      0.00         0

    accuracy                           0.86       191
   macro avg       0.57      0.55      0.56       191
weighted avg       0.86      0.86      0.86       191
```



ABSA Model Exploration

Multi-Aspect

Models Explored

- BERT for Sequence Classification Model
- Llama 3.1
- Facebook BART
- DeBERTa

Aspect-Based Sentiment Analysis

BERT for Sequence Classification Model

	Bert for Sequence Classification
Technique	Leverage on the knowledge of the pre-trained model without any fine-tuning
Labels	Positive, Negative, Neutral
Observation	• Incorrect output

Sample Output :

```
# Example usage
review = "Fast ship good price. It's a shame I had to order this from another walmart and not my local store."
aspects = ['Price', 'Delivery']
```

```
[Price] [sentiment: negative]
[Delivery] [sentiment: negative]
```

! Missed sentiment nuances

Aspect-Based Sentiment Analysis

Llama 3.1

	Llama 3.1
Technique	Zero-Shot
Prompts	What is the sentiment (positive, negative) for this aspect? Return the sentiment identified only.
Observation	• Impressive results ✔ • Consistent output format

Sample Output :

```python
def extract_sentiment_expression_llama(review, aspects, model_name='llama3.1'):
    # Store the answers
    answers = {}

    # Iterate over the provided aspects to construct the prompt
    for aspect in aspects:
        prompt = f"""
        Review: "{review}"
        Aspect: "{aspect}"
        What is the sentiment (positive, negative) for this aspect? Return the sentiment identified only.
        """

        # Use the Ollama API to generate the sentiment expression
        response = ollama.chat(
            model=model_name,
            messages=[{"role": "user", "content": prompt}]
        )

        # Extract the sentiment expression from the response
        result_text = response['message']['content']
        answers[aspect] = result_text.strip()

    return answers
```

Aspect-Based Sentiment Analysis

Facebook BART

	Facebook BART
Technique	Zero-Shot
Prompts	What is the sentiment (positive, negative) for this aspect? Return the sentiment identified only.
Observation	• Impressive results ✔ • Consistent output format

Sample Output :

```
# Function to extract sentiment expressions
def extract_sentiment_expression_nli(review, aspects, nli_pipeline):
    # Define possible labels for sentiment
    candidate_labels = ['positive', 'negative']

    # Store the answers
    answers = {}

    # Iterate over the provided aspects to construct the NLI inputs
    for aspect in aspects:
        # Formulate the hypothesis
        hypothesis = f"The sentiment for the aspect '{aspect}' is"

        # Use the NLI pipeline to predict the sentiment for each aspect
        response = nli_pipeline(
            sequences=review,    # Premise: The review text
            candidate_labels=[f"{hypothesis} {label}" for label in candidate_labels],  # Hypotheses
        )

        # Extract the sentiment with the highest score
        sentiment = response['labels'][0].split()[-1]  # Get the last word ('positive' or 'negative')
        answers[aspect] = sentiment

    return answers

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

Aspect-Based Sentiment Analysis

DeBERTa

	DeBERTa
Technique	Zero-Shot
Prompts	What is the sentiment (positive, negative) for this aspect? Return the sentiment identified only.
Observation	• Impressive results ✔ • Consistent output format

Sample Output :

```
# Example usage
review = "Fast ship good price. It's a shame I had to order this from another walmart and not my local store."
aspects = ['Price', 'Delivery']
Sentiment Expressions: {'Price': 'Positive', 'Delivery': 'Positive'}
```

Summary output for Mar-24 and Apr-24 reviews

			Llama3.1	
	Data	**Sentiment**	**Positive**	**Negative**
Metrics:	**Mar'24**	Accuracy Rate	79.6%	79.6%
		Precision Rate	97.5%	71.4%
		Recall Rate	61.1%	98.4%
	Apr'24	Accuracy Rate	78.8%	78.8%
		Precision Rate	97.1%	70.2%
		Recall Rate	60.7%	98.1%

Low Recall Rate for Positive Sentiments

Conclusion & Final Model

Performance Evaluation

Performance Evaluation

Modular Systems

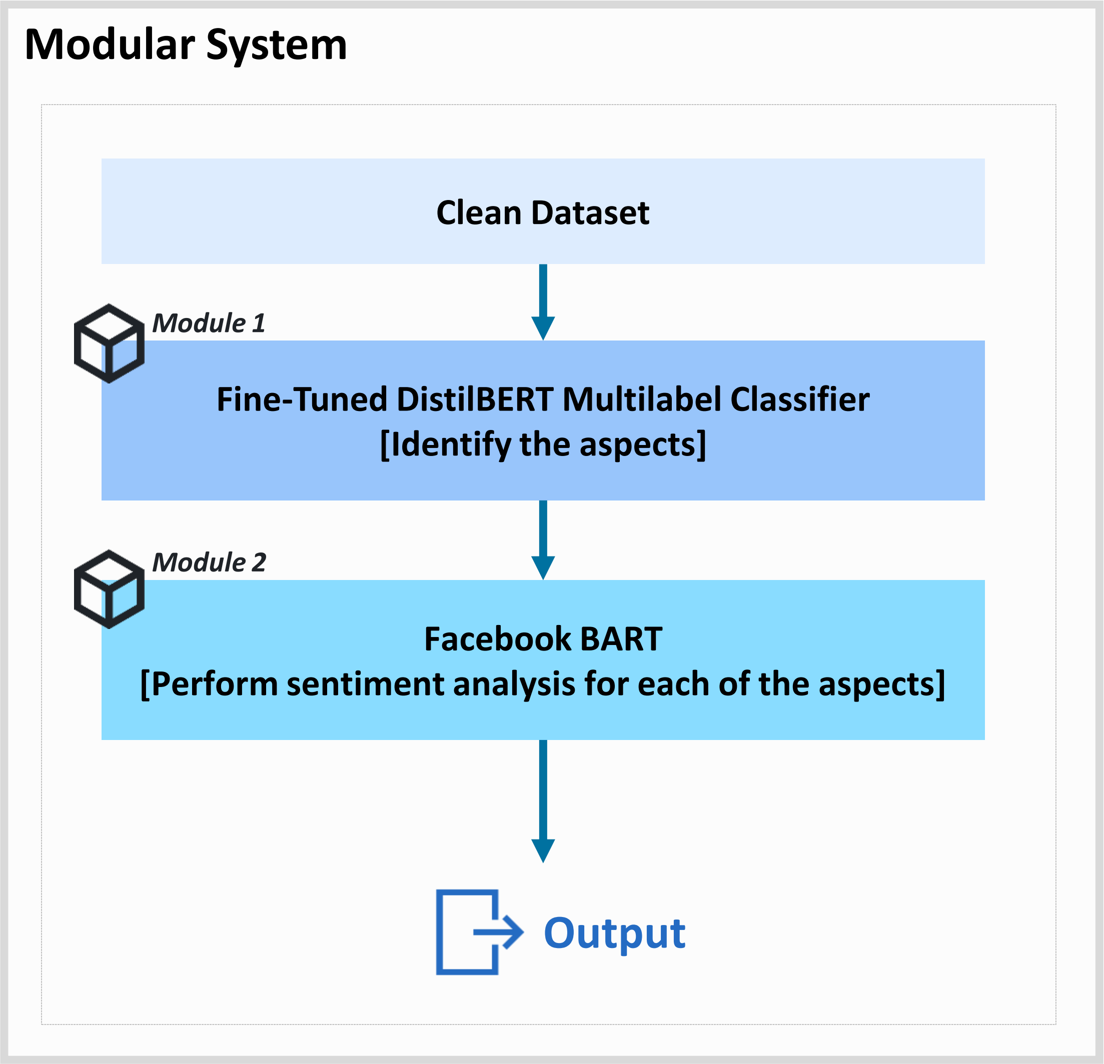
Aspect-Based Sentiment Analysis

		Lexicon-Based				Classical Machine Learning				Zero-Shot		Natural Language Inference			
		TextBlob		Vader		Naïve Bayer		SVM		Llama3.1		Facebook BART large mnli		DeBERTa-v3-base-mnli-fever-anli	
Data	**Sentiment**	**Positive**	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**
Mar'24	Accuracy Rate	77.9%	77.9%	77.7%	77.7%	76.1%	76.1%	76.9%	76.9%	79.6%	79.6%	81.6%	81.6%	80.3%	80.3%
	Precision Rate	82.9%	66.5%	86.9%	63.0%	77.8%	69.7%	82.1%	65.0%	97.5%	71.4%	87.3%	77.4%	87.5%	75.4%
	Recall Rate	84.9%	63.1%	79.0%	75.0%	90.5%	45.8%	84.3%	61.4%	61.1%	98.4%	74.3%	89.1%	71.0%	89.7%
Apr'24	Accuracy Rate	77.8%	77.8%	80.8%	80.8%	75.8%	75.8%	76.5%	76.5%	78.8%	78.8%	81.5%	81.5%	80.1%	80.1%
	Precision Rate	82.7%	66.3%	89.3%	66.8%	78.3%	67.2%	83.0%	62.9%	97.1%	70.2%	87.9%	76.6%	87.8%	74.6%
	Recall Rate	85.3%	61.8%	81.7%	79.1%	89.2%	47.1%	82.4%	63.9%	60.7%	98.1%	74.3%	89.1%	71.3%	89.5%

Note: The diagram above displays the performance for reviews containing a **single aspect**, as there was no labeled data available for sentiment analysis across multiple aspects.

Final Model

DistilBERT Multilabel + FBBart



System Demo

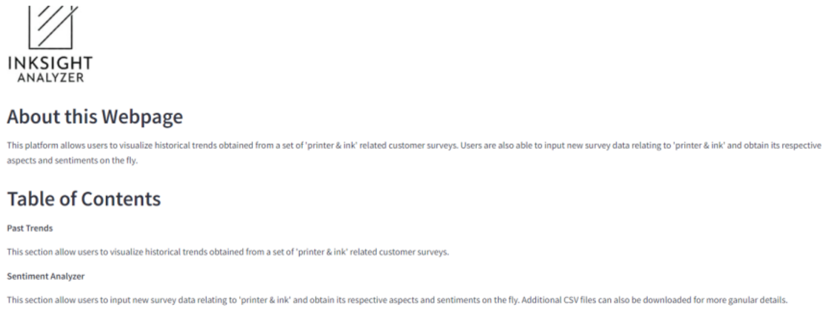
Streamlit Interface

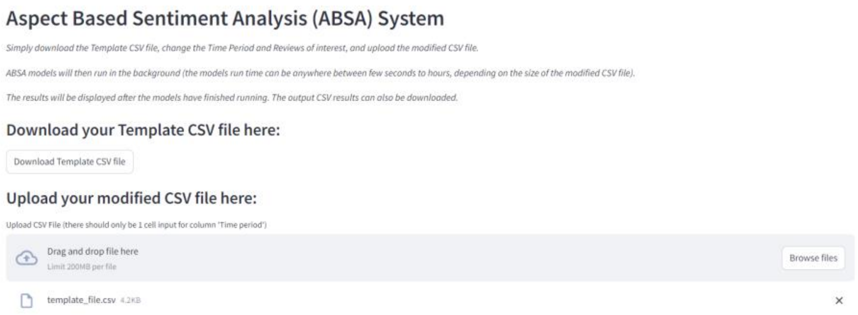
Frontend UI & Backend Codes

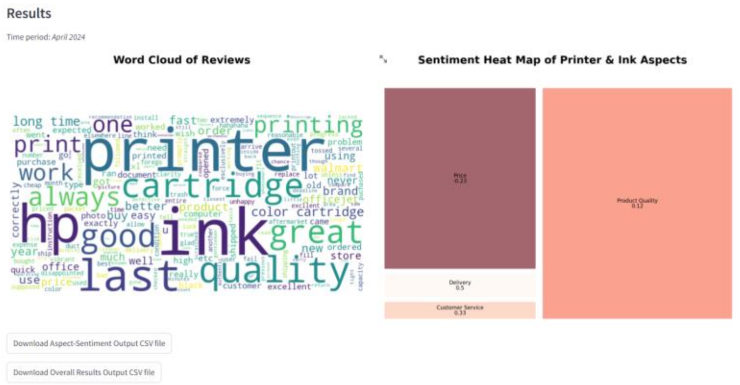
- System Demonstration
- Data Pipeline

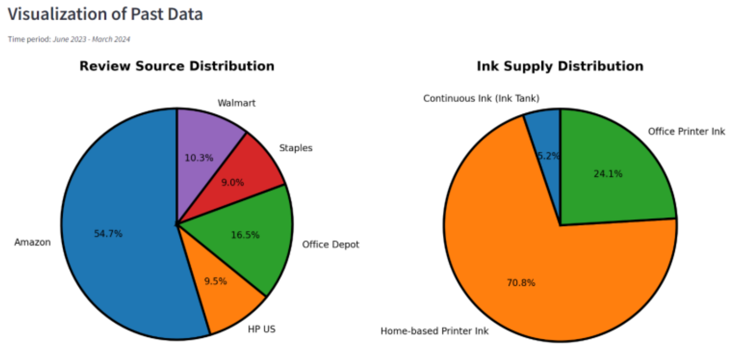
System Demo

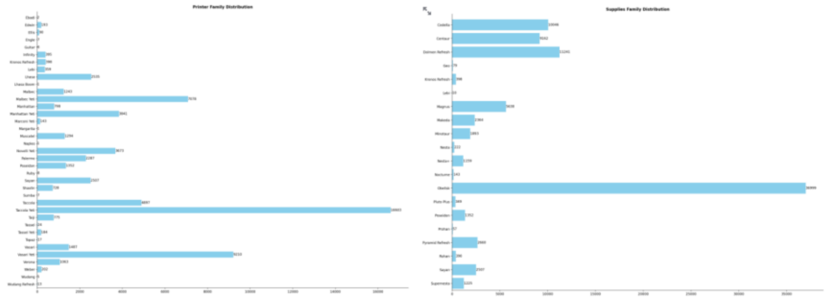
- Front-end UI demonstration via Streamlit (https://inksightanalyzer.streamlit.app)

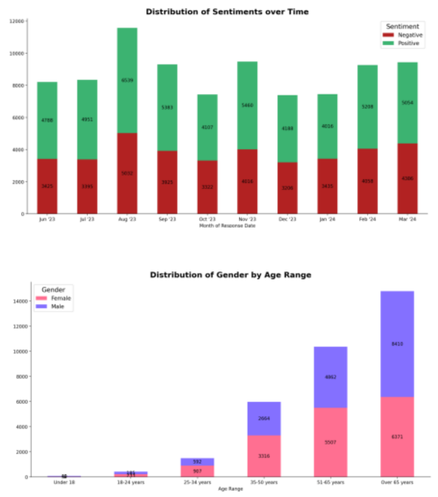












Data Pipeline

How does it work???

- ControllerService Class
  - AspectClassification()
  - SentimentAnalyzer()

Model running codes

- Streamlit Pages
  - 🏠_Homepage.py
  - 1_Past Trends.py
  - 2_Sentiment Analyzer.py

Frontend UI

Data Pipeline (Model codes)

- AspectClassification()
  - Run classification model per review
  - Model returns single/ multi-aspects as outputs (Price, Customer Service, Product Quality, Delivery)
  - Data post-processing
    - May split each review into multi-rows, based on number of aspects extracted
  - Function returns 1 dataframe:
    - **Single/ multiple aspects per review**

```python
def aspectClassification(self, rawInput_file):

    # Aspects you're classifying
    aspects = ['Price', 'Customer Service', 'Product Quality', 'Delivery']

    def classify_aspect(row, aspects):
        # Classify aspect using the provided classifier
        aspect_result = classifier(row, aspects)
        aspect = aspect_result['labels'][0]  # The most probable aspect
        return aspect
```

AspectClassification function

Data Pipeline (Model codes)

- SentimentAnalyzer()
  - Run sentiment model per aspect, per review
  - Model returns either “Positive” or “Negative” as outputs
  - Function returns 2 dataframes:
    - **Aspect-Sentiment pairs per review**
    - **Aspects’ counts & sentiment scores**

```python
def sentimentAnalyzer(self, aspectInput_df):

    df = aspectInput_df.copy()


    def classify_sentiment(row):
        # Classify sentiment using the sentiment model
        sentiment_result = sentiment_model(row)
        sentiment = sentiment_result[0]['label']
        return sentiment

    # Apply sentiment classification and append the result
    df['Sentiment'] = df['Sentence'].apply(classify_sentiment)

    df.loc[df.Sentiment == 'POSITIVE', 'Sentiment'] = 'Positive'
    df.loc[df.Sentiment == 'NEGATIVE', 'Sentiment'] = 'Negative'
    aspectSentimentOutput_df = df.copy()
```

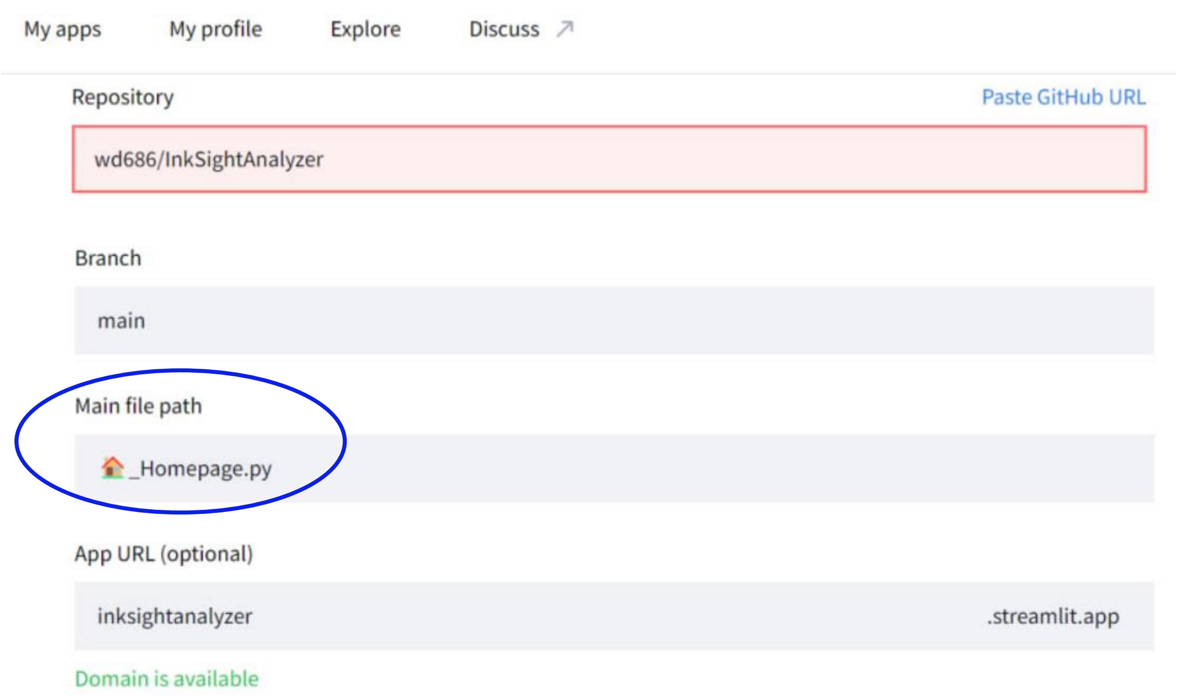
SentimentAnalyzer function

```python
class controllerService:

    def __init__(self):
        return

    def runAspectClassification(self, rawInput_file):
        return aspectClassification(self, rawInput_file)

    def runsentimentAnalyzer(self, aspectInput_df):
        return sentimentAnalyzer(self, aspectInput_df)
```

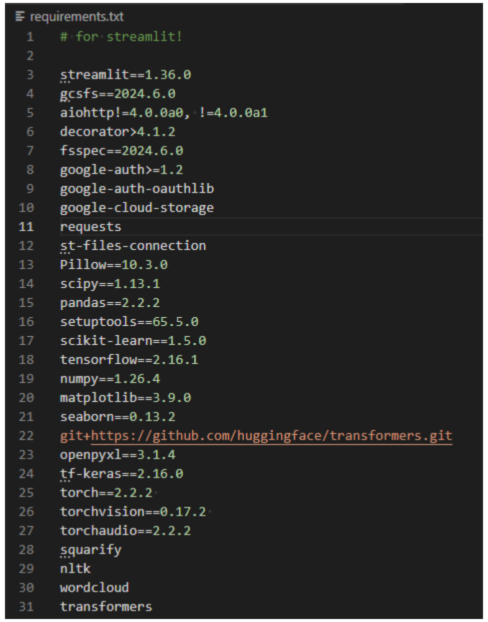
ControllerService class to hold both functions

Data Pipeline (Streamlit)

- Create Streamlit webpage & link to GitHub repository
- Prepare Requirements.txt
    - Installs required packages on server



Create Streamlit webpage



Requirements.txt

Data Pipeline (Streamlit – Past Trends)

- Bottleneck
  - High latency when running data source file (101.2MB) to generate visualizations on the fly
- Solution
  - Pre-run codes to generate smaller dataframes (< 2.5MB) needed for visualization (reduces page loading time)

```python
# reviewSource pie chart df
df2.loc[(df2['Review Source'].notnull()) & (df2['Review Source'].str.contains('amazon', case = False)), 'Review Source'] = 'Amazon'
reviewSource_df = df2.groupby('Review Source').count().reset_index()

# inkSupply pie chart df
inkSupply_df = df2.groupby('Ink Supply Type').count().reset_index()

# printer family bar chart df
df2['Printer Family'] = df2['Printer Family'].str.strip().str.title()
printer_df = df2.groupby('Printer Family').count().sort_values(ascending = False, by = 'Printer Family').reset_index()

# supplies family bar chart df
df2['Supplies Family'] = df2['Supplies Family'].str.strip().str.title()
supplies_df = df2.groupby('Supplies Family').count().sort_values(ascending = False, by = 'Supplies Family').reset_index()

# age/ gender stacked bar chart df
ageGender_df = df2[(df2['Age Range'].notnull()) & (df2['Gender'].notnull())][['Age Range', 'Gender']].reset_index(drop = True)
ageGender_df = ageGender_df[((ageGender_df.Gender == 'Male') | (ageGender_df.Gender == 'Female')) & (~(ageGender_df['Age Range'] == 'Prefer not to answer'))]

# sentiment/ time stacked bar chart df
def score_to_sentiment(row):
    if not pd.isna(row['LTR']):
        # Use LTR (0-10)
        if row['LTR'] <= 6:
            return 'Negative'
        else:
            return 'Positive'
    elif not pd.isna(row['Star Rating']):
        # Use Star Rating (1-5)
        if row['Star Rating'] <= 3:
            return 'Negative'
        else:
            return 'Positive'
    else:
        return 'Unknown'
sentiment_list = []
for index, row in df2.iterrows():
    sentiment_list.append(score_to_sentiment(row))
df2['sentiment'] = sentiment_list
sentimentTime_df = df2[df2.sentiment.isin(['Negative', 'Positive'])][['sentiment', 'Month of Response Date']]
```

Dataframes transformation for visualization

```python
startEndPeriods_df = pd.read_csv('Sandbox/streamlitProcessing/generatedCSVs/startEndPeriods.csv')
reviewSource_df = pd.read_csv('Sandbox/streamlitProcessing/generatedCSVs/reviewSource.csv')
inkSupply_df = pd.read_csv('Sandbox/streamlitProcessing/generatedCSVs/inkSupply.csv')
printer_df = pd.read_csv('Sandbox/streamlitProcessing/generatedCSVs/printer.csv')
supplies_df = pd.read_csv('Sandbox/streamlitProcessing/generatedCSVs/supplies.csv')
ageGender_df = pd.read_csv('Sandbox/streamlitProcessing/generatedCSVs/ageGender.csv')
sentimentTime_df = pd.read_csv('Sandbox/streamlitProcessing/generatedCSVs/sentimentTime.csv')
```

Pre-generate dataframes as CSVs & load for visualization

Data Pipeline (Streamlit – Sentiment Analyzer)

- Instantiates ControllerService class
- User input’s survey data will be fed into pipeline:
  - AspectClassification() → SentimentAnalyzer() → Other Streamlit codes (visualization, file downloads)
- AspectClassification() & SentimentAnalyzer() functions will run to execute the Aspect Based Sentiment Analysis (ABSA) models
- Word Cloud and Tree Map will be generated
- 2 dataframes generated from SentimentAnalyzer() function will be made available for download

- SentimentAnalyzer()
  - Run sentiment model per aspect, per review
  - Model returns either “Positive” or “Negative” as outputs
  - Function returns 2 dataframes:
    - **Aspect-Sentiment pairs per review**
    - **Aspects’ counts & sentiment scores**

(From earlier slide ..)

```
st.download_button("Download Aspect-Sentiment Output CSV file",
                   aspectSentimentOutput_df.to_csv(index = False),
                   file_name = 'aspectSentimentOutput_file.csv',
                   mime = 'text/csv')

st.download_button("Download Overall Results Output CSV file",
                   overallResultsOutput_df.to_csv(index = False),
                   file_name = 'overallResultsOutput_file.csv',
                   mime = 'text/csv')
```

Create download buttons for 2 output dataframes

Conclusion

Challenges, Limitations and Restrospective

Challenges

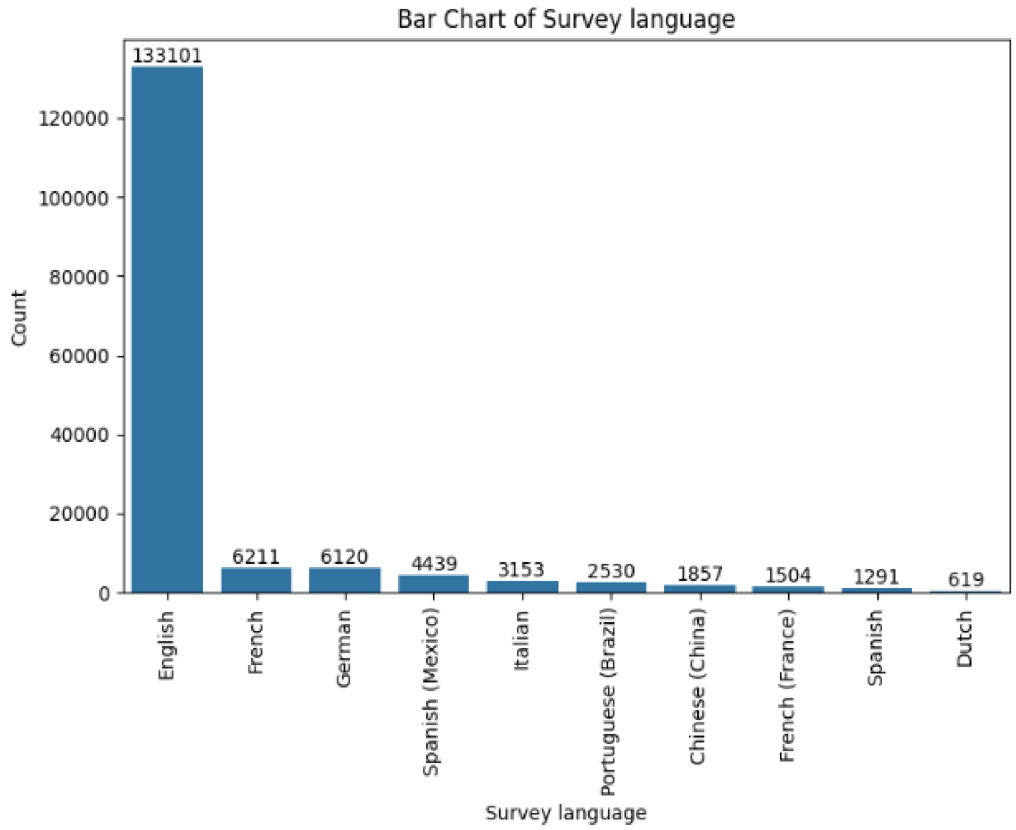
- **Inconsistent LTR and Star Rating Score versus the sentiment**
  - Due to assumption that LTR and Star rating are proxies of one another
  - Manual labelling of categories in line with business requirements

- **Ambiguous review**
  - "Good but expensive but this is Biden' fault"
  - Inability to correctly group some reviews into "neutral" sentiment, considered exclusion or prioritise relabelling into positive reviews

- **Print Quality vs Printer Quality**
  - “The printer does produce good quality printing however the ink is very expensive and it doesn't last very long. There are some cheaper inks available, however HP does not recommend them.”

Limitations

- English reviews, Spanish or French for translation, losing semantic or linguistic nuances
- Reviews for multiple product purchase not distinguishable between products (currently assumed all to be ‘Product Quality’)
- Lack data on labels for multiple aspects for modelling
- Sentiment modelling was sufficient therefore does not require finetuning component
- Discrepancy for scoring sentiment by manual labelling
  - Taking LTR score based on feedback from reviews



Conclusion

Considerations for future projects

- **DistilBERT is the best multilabel classifier for aspect identification, where the aspects identified will be loaded onto NLI (Facebook BART) as it outperforms other models for sentiment extraction.**
- Start earlier with well-established baseline models (e.g., Naive Bayes, SVM, Random Forest) to establish a benchmark for performance evaluation
- Incorporate other deep learning architectures like recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformers for more complex tasks and potentially better performance
- Utilise hybrid approaches: Combine traditional machine learning techniques with deep learning models to leverage their respective strengths


keep reinventing