

# PROJECT REPORT

PRACTICE MODULE FOR CERTIFICATE IN:

Practical Language Processing (PLP)



---

## Boosting Customer Satisfaction via Practical Language Processing

Using Bank Call Centre Customer Dialog Data

---

Team Members:

Richard Chai

## Contents

1.0 Executive Summary .....	4
2.0 Problem Statements.....	4
2.1 Manual classification and tabulation of Customer Satisfaction is error prone and not scalable .....	4
2.2 Taking impactful action to increase Customer Satisfaction is difficult without knowing specific areas to address .....	5
3.0 Solution and Scope .....	6
3.1 Solution.....	6
3.1.1 Aspect Topic – Sentiment Analysis .....	7
3.1.2 Latent Topics – Sentiment Analysis.....	7
3.1.3 Running the Dashboard Program .....	7
3.2 Scope.....	9
3.3 Model Building Process .....	10
3.3.1 Assembling the Unlabelled CCD Dataset.....	10
3.3.2 Dataset Cleaning and Preparation .....	11
3.3.3 Automated Data Labeling using LLM.....	11
3.3.4 Generate Synthetic Multi-Turn Dialog.....	12
3.3.5 The Final Labelled CCD.....	12
3.3.6 Train final classification models .....	13
3.4 Components of Automated Topic Sentiment Analysis Solution .....	16
3.4.1 Data Cleaning and Preparation.....	16
3.4.2 Aspect Topic Module .....	16
3.4.3 Aspect Topic Accuracy Modelling .....	17
3.4.4 Topic Modelling Module .....	17
3.4.5 Sentiment Classification Calculation .....	20
3.4.6 Sentiment Accuracy Monitoring .....	21
3.4.7 Topics Database .....	22
3.4.8 Customer Satisfaction Metrics.....	22
3.4.9 Dashboard .....	25
3.4.10 Conversation User Interface.....	27
4.0 System Architecture .....	27
5.0 Conclusion / Area of Improvement .....	28
6.0 Bibliography & Appendix .....	29
6.1 Appendix A - Project Files .....	29
6.2 Appendix B – Datasets Used .....	29

6.3 Appendix C - Citations.....	30
6.4 Appendix D: Usage of ChatGPT .....	31

## 1.0 Executive Summary

Over the last few years, the bank has been executing its strategy of customer centricity and it has been rewarded with excellent financial results and industry accolades. To monitor the success of this strategy, the bank utilises 3<sup>rd</sup> party agencies and internal staff to monitor customer satisfaction levels.

The 3<sup>rd</sup> party agencies claim to have highly effective proprietary methods which the bank does not have access to. Within the bank, the categorization and tabulation of customer satisfaction levels is performed manually, with results documented on Excel spreadsheets. This laborious process is error-prone and hinders the generation of timely customer satisfaction updates, typically limited to a quarterly frequency.

Hence, management would like to try using Practical Language Processing (PLP) techniques on customer dialog data to:

- Automate the manual process of monitoring customer satisfaction levels.
- Identify areas where the bank is doing well or not, with system recommended prioritisation.
- Validate the accuracy of the existing customer satisfaction monitoring services.

By implementing this Customer Dialog Analysis System, the bank will be able to take a data-driven approach to gain deep insights into their customer interactions, allowing them to proactively address emerging issues and adapt quickly to changing customer needs and expectations.

## 2.0 Problem Statements

### 2.1 Manual classification and tabulation of Customer Satisfaction is error prone and not scalable

The bank stores customer dialog for compliance and training purposes and the volume is huge. This makes manual analysis of the customer dialog data very laborious.

Moreover, the dialog may contain ambiguity. For example,

- The staff serving me was excellent, very polite and extremely attentive, but the solution provided could be better.
- You have the worst savings interest rate among all the banks, and the only thing stopping me from switching banks is the short waiting times and the excellent customer service I receive every time I visit.

For both example dialog, should they be tagged with a sentiment of positive or negative, or neutral?

Even with detailed instructions and criteria, it is likely that different human labellers will label them differently. Not only that, but it is also possible that the same human labeller, if asked to label the same dialog more than once (after an interval), would label the sentiment polarity inconsistently.

By using Practical Language Processing (PLP) techniques to create AI models we can perform labelling of customer dialog in a more consistent and scalable manner.

## 2.2 Taking impactful action to increase Customer Satisfaction is difficult without knowing specific areas to address

Merely knowing if the customer is satisfied or dissatisfied with the bank is insufficient if we want to take effective action to improve the situation.

To do this, we need insights that allow us to target precisely for the greatest impact.

Hence, we will create a system that performs **Aspect Based Sentiment Analysis (ABSA)** which provides granular insights of customer satisfaction of specific products and customer usage.

From the results of the ABSA, we will derive the following:

- Net Promoter Score (NPS)
- Customer Satisfaction Score (CSAT)

With such granular insights, the bank can identify areas where customers are satisfied or dissatisfied and can then create targeted actions to:

- Improve Customer Experience
  - By understanding where customers are unhappy, intervention initiatives can be created to address pain points and enhance the overall customer experience.
  - This could involve making product improvements, providing better support, or offering incentives to encourage continued use.
- Promote Successful Product Usage
  - Insights into areas where customers are successfully using products and services can be helpful in driving the bank's marketing and educational efforts.
  - The bank can highlight these positive use cases to provide guidance to help other customers get the most out of the products, and potentially cross-sell related offerings.
- Optimize Product Mix
  - Analysing customer satisfaction and usage trends across the bank's product portfolio enables data-driven decisions about which products to invest in, maintain, or phase out.
  - This optimization helps the bank allocate resources efficiently and ensure its offerings align with customer needs and preferences.

By applying Practical Language Processing to the bank's customer dialog data, we can help the bank take precise and impactful data driven actions to boost customer satisfaction, be more responsive to customer needs and deliver greater value in a scalable manner.

## 3.0 Solution and Scope

### 3.1 Solution

Two problem statements were highlighted in section 2.0

- Manual classification and tabulation of Customer Satisfaction is error prone and not scalable
- Taking impactful action to increase Customer Satisfaction is difficult without knowing specific areas to address

The proposed solution is to create **Automated Customer Dialog Analysis System** that determines the sentiment polarity of *Aspect Topics* and *Topic Modelling (Latent Topics)* without the need for tedious manual work.

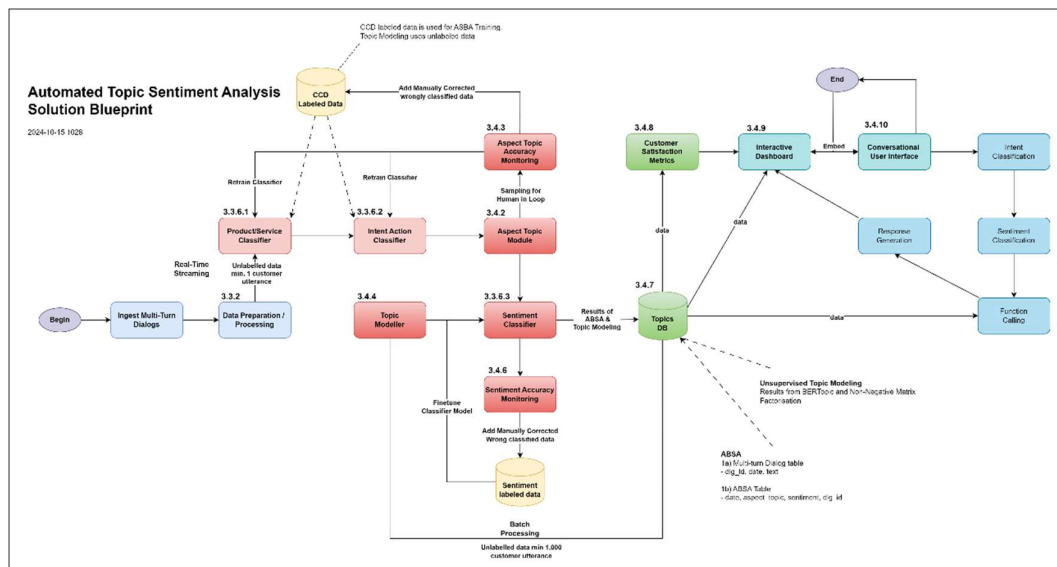


Figure 3.1.1 Solution Blueprint – Automated Topic Sentiment Analysis System

Aspect Topic focuses on how the customer was using a particular product or service while Latent Topics may uncover topics that are outside the scope of Aspect Topics. Together, they enable the bank to create intervention action plans that are tightly focused to achieve greater impact.

For example, if a customer said that “your branch service was amazing, but the choice of investment plans is terrible.”

An Aspect Based Sentiment Analysis (ABSA) should give a result as follows:

- Aspect Topic: Branch Service / Sentiment: Very Positive
- Aspect Topic: Choice of Investment Plans / Sentiment: Negative

However, Aspect Topics need to be defined beforehand or created using a rule-based approach during runtime.

Hence, the solution also provides a second method, Topic Modelling, to uncover latent topics.

To summarise, the system

- Identifies topics from customer dialog
- Identifies the product and usage areas the bank is doing well (or not)
- Prioritises them based on a relevant importance and/or sentiment metric so that the bank can make a data-driven decision on intervention initiatives.
- Also provides bank wide metrics like Net Promoter Score (NPS) and Customer Satisfaction Ratio (CSAT) for the bank to have an overall sense of their standing in their customers' eyes.

### 3.1.1 Aspect Topic – Sentiment Analysis

Aspect Topic is defined as “customer action (intent) + product/solution”

Examples

Aspect_Topic (action + __ + product/solution )	I want to	Action (intent)	Product	Description
close__account	I want to 'close__account'	close	account	The customer wants to close an unknown type of account.
deposit__account	I want to 'deposit__account'	deposit	account	The customer wants to make a deposit to an unknown type of account.
feedback__savings_account	I want to 'feedback__savings_account'	feedback	Savings_account	The customer wants to provide feedback about savings account. This Aspect Topic is also used as a fallback aspect.

Sentiment analysis is applied on each Aspect Topic which can be used to prioritise the bank's resources and efforts in improving customer satisfaction.

### 3.1.2 Latent Topics – Sentiment Analysis

In addition to Aspect Topic, the system has a second mode which performs unsupervised Topic Modelling to discover latent topics.

Latent topics are underlying themes within the multi-turn dialog which are uncovered through topic modelling. They are inferred from patterns of word co-occurrences, rather than explicit labels.

Sentiment analysis can be applied to these latent topics to identify areas of focus which the bank can drill down to apply intervention activities to improve customer satisfaction.

More details are found in 3.4.4.

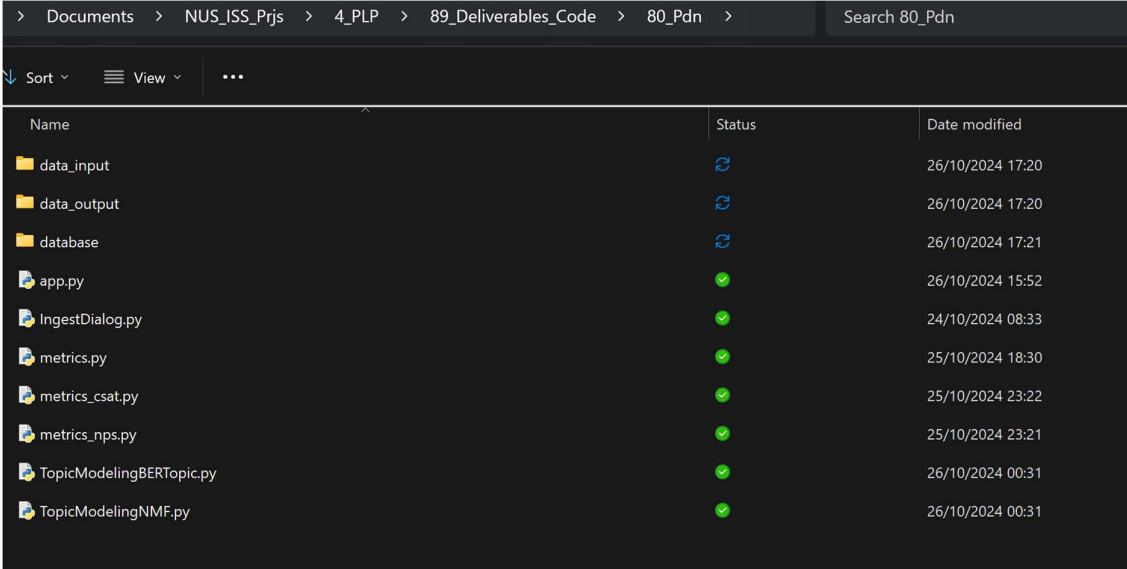
### 3.1.3 Running the Dashboard Program

Ensure that you have the following requirements installed:

- pypjupyterlab 4.2.5
- ipywidgets 8.1.5

- torch 2.4.1
- datasets 3.0.0
- transformers 4.44.2
- accelerate 0.34.2
- gradio 4.44.0
- evaluate 0.4.3
- zstandard 0.23.0
- psutil 6.0.0
- soundfile 0.12.1
- librosa 0.10.2.post1
- bertopic 0.16.3
- textblob 0.18.0.post0
- openai 1.51.2
- python-dotenv 1.0.1
- openpyxl 3.1.5
- duckdb 1.1.2
- seaborn 0.13.2
- streamlit 1.39.0

Next,



Name	Status	Date modified
data_input		26/10/2024 17:20
data_output		26/10/2024 17:20
database		26/10/2024 17:21
app.py		26/10/2024 15:52
IngestDialog.py		24/10/2024 08:33
metrics.py		25/10/2024 18:30
metrics_csat.py		25/10/2024 23:22
metrics_nps.py		25/10/2024 23:21
TopicModelingBERTopic.py		26/10/2024 00:31
TopicModelingNMF.py		26/10/2024 00:31

- Navigate to the folder containing the python files as seen in the above screenshot. If you did not rename the folders, it should be in “80\_pdn” folder.

Start your virtual environment if you wish to.

Then run the command:

- streamlit run app.py

And you should see a screen like this:





### 3.2 Scope

For this Proof-of-concept (POC), our goal is to create a minimum viable project (MVP) within 2 sprints and gather user feedback for further development.

As time is limited, we will limit the scope of Aspect Based Sentiment Analysis (ABSA) in this POC to the following product and service areas:

The current list of products and actions we are monitoring include:

Product List	Action List
savings account	apply
current account	access
account	activate
loan	block
fixed deposit	cancel
forex account	close
mobile app	deposit
website	dispute
atm	earn
credit card	exchange
debit card	find
card	inquire
others	link
Not Used	open
	pay
	receive
	redeem
	refund
	renew
	report
	reset
	retrieve
	schedule
	select
	transfer
	unblock

	update
	unlink
	verify
	withdraw
	Unknown

More products and actions which form the aspect topic can be added in future iterations to broaden the scope of Aspect Based Sentiment Analysis.

As for topic modelling, there are no limitations on topics because what is uncovered depends on the input data.

### 3.3 Model Building Process

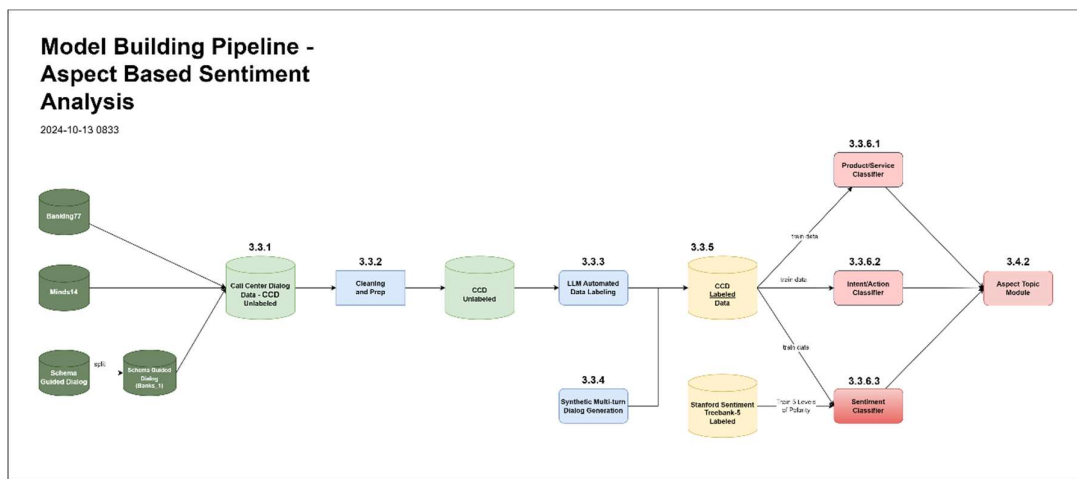


Figure 3.3 Model Building

Because there are not many banking dialog related datasets available publicly, the dataset preparation is complex.

In addition to data cleaning and preparation, many of the datasets are not annotated with the intents, entities and sentiment polarity that is required for this project.

Manual labelling is not scalable and error-prone; hence an automated approach is taken, with human-in-the-loop to obtain a fully annotated dataset which is used to train the final classification models. More details below.

#### 3.3.1 Assembling the Unlabelled CCD Dataset

Datasets Used:

- Banking77
  - This dataset is composed of online banking queries annotated with their corresponding intents.
  - <https://huggingface.co/datasets/PolyAI/banking77>
- Minds14

- This dataset is a training and evaluation resource for intent detection task with spoken data, also has data related to e-banking.
- <https://huggingface.co/datasets/PolyAI/minds14>
- Schema-Guided Dialog (SGD) DSTC8
  - The Schema-Guided Dialogue dataset (SGD) was created for the Dialogue State Tracking Challenge (DSTC8), focusing on multi-domain, task-oriented conversation, has some banking related dialog.
  - [https://huggingface.co/datasets/google-research-datasets/schema\\_guided\\_dstc8](https://huggingface.co/datasets/google-research-datasets/schema_guided_dstc8)

As the available datasets for the banking domain are few, the above datasets will be combined to form the *Call Centre Dialog* dataset aka (**CCD**).

Jupyter Notebook

- Create Call Centre Dialog CCD Unlabeled dataset.ipynb

### 3.3.2 Dataset Cleaning and Preparation

#### 3.3.2.1 Data Preparation for Aspect Based Sentiment Analysis

Steps:

1. As we are only interested in discovering topics on what the customer wants, we keep only the customer's utterance.
2. Dialog paragraphs are split into sentences.
3. Sentences that are less 2 words or less are dropped.
4. Sentences that have no relationship with the bank's products are dropped.

Jupyter Notebook

- Create Call Centre Dialog CCD Unlabeled dataset.ipynb
- Create Call Centre Dialog CCD Unlabeled dataset.html

#### 3.3.2.2 Data Preparation for Topic Modelling

Currently data preparation for topic modelling is the same as for ABSA (3.3.2.1).

### 3.3.3 Automated Data Labeling using LLM

An LLM (GPT-4o) is used to create labels for the unannotated CCD dataset. This dataset will be used to train the product, action and sentiment classification models.

*Example:*

	date	dlg_id	text	product	action	sentiment
0	2024-10-05	2024-10-07_1017--0	i am still waiting on my card	card	inquire	negative
1	2024-10-05	2024-10-07_1017--1	what can i do if my card still hasnt arrived a...	card	inquire	negative
2	2024-10-05	2024-10-07_1017--2	i have been waiting over a week is the card st...	card	inquire	negative
3	2024-10-05	2024-10-07_1017--3	can i track my card while it is in the process...	card	inquire	neutral
4	2024-10-05	2024-10-07_1017--4	how do i know if i will get my card or if it i...	card	inquire	neutral

As the LLM does not always follow instructions precisely, some manual adjustments is still required to correct the labels.

Jupyter Notebook:

- Use\_LLM\_to\_Label\_Data\_v02.ipynb

### 3.3.4 Generate Synthetic Multi-Turn Dialog

As the number of dialogs is still insufficient, GPT-4o was used to generate another 2,000 bank related multi-turn dialog.

1,500 dialogs are added to the labelled CCD in 3.3.3 to form the final labelled CCD in 3.3.5

500 dialogs will be used as unseen data to test the final classification models created in 3.3.6

Stratified random sampling is used to split the data.

Jupyter Notebooks:

- Gen\_Synthetic\_Multi\_Turn\_Dialog\_v05.ipynb

### 3.3.5 The Final Labelled CCD

Example of the final labelled customer centre dialog dataset (CCD) (below) used for training classification models.

Date	Text	product	action/intent	sentiment
2024-10-02	I have been waiting a long time for my credit card to be activated	card credit	activate	negative
2024-10-02	This is ridiculous! My credit card has not arrived in my mail even after a month	card credit	activate	very negative
2024-10-02	With the integration to WeChat, making payments for my monthly credit card bills is now a breeze	card credit	pay	positive
2024-10-03	The loan interest rate is the best among all the banks	loan	feedback	very positive
2024-10-03	Tell me about the promotion rates for opening a new fixed deposit	fixed deposit	redeem	Negative

Jupyter Notebook:

- Split and Create Datasets.ipynb

### 3.3.6 Train final classification models

#### 3.3.6.1 Product/Service Classifier

The purpose of the Product/Service Classifier is to determine which bank product the customer is referring to during the conversation with the Customer Service Executive or Chatbot.

<b>Text</b> (input)	<b>Product/Service</b> (output of this classifier)
The loan interest rate is really good,	loan
but closing fixed deposit takes too long.	fixed_deposit

The list of products enabled for this POC is found in 3.2 (Scope)

#### Trained/Fine-tuned Models

<b>Model</b>	<b>Accuracy</b>	<b>Weighted Ave F1</b>	<b>Test Runtime</b>
Logistic Regression	0.77	0.69	-
Multinomial Naive Bayes (MNB)	0.56	0.37	-
BERT-base	0.80	0.80	12.23
distilBERT	0.79	0.79	6.52
mobileBERT	0.79	0.79	10.7
tinyBERT	0.75	0.74	1.69

For POC, selected tinyBERT for fastest run-time with small drop in accuracy and f1. If overall system results are not idea, then switch to distilBERT. Due to scarce GPU, BERT models were only trained for 3 epochs. Further training may result in better performance.

#### Jupyter notebook

- Pdt\_Clf\_Logistic\_Regression.ipynb
- Pdt\_Clf\_MNB.ipynb
- pdt\_clr\_BERT\_Multiple.ipynb

#### 3.3.5.2 Action/Intent Classifier

The purpose of the Action/Intent Classifier is to determine the “action” portion of the customer’s intent during the conversation with the Customer Service Executive or Chatbot.

<b>Text</b> (input)	<b>Product/Service</b> (input)	<b>Product/Service</b> (output of this classifier)
The loan interest rate is really good,	loan	feedback
but closing fixed deposit takes too long.	fixed_deposit	redeem

The list of products enabled for this POC is found in 3.2 (Scope)

#### Trained/Fine-tuned Models

<b>Model</b>	<b>Accuracy</b>	<b>Weighted Ave F1</b>	<b>Test Runtime</b>
Logistic Regression	0.57	0.42	-
Multinomial Naive Bayes (MNB)	0.31	0.09	-
BERT-base	0.67	0.67	11.17
distilBERT	0.66	0.66	5.82
mobileBERT	0.65	0.65	10.23
tinyBERT	0.42	0.35	1.57

For POC, selected distilBERT. Although tinyBERT is much faster than distilBERT but the difference in accuracy is too large to ignore. Due to scarce GPU, BERT models were only trained for 3 epochs. Further training may result in better performance.

#### Jupyter notebook

- action\_Clf\_Logistic\_Regression.ipynb
- action\_Clf\_MNB.ipynb
- action\_clr\_BERT\_Multiple.ipynb

#### 3.3.6.3 Sentiment Classifier

The standard Hugging Face sentiment analysis models typically classify sentiments as either positive or negative.

To have 5 levels of sentiment polarity: [“very negative”, “negative”, “neutral”, “positive”, “very positive”], I aggregated existing dataset and created synthetic data to train models for this purpose.

#### Example

<b>Text</b> (input)	<b>Sentiment (5 levels)</b> (output of this classifier)
The loan interest rate is the best in the market.	very positive
closing fixed deposit takes too long.	negative

#### Dataset:

- Stanford Sentiment Treebank
  - with 5 labels: very positive, positive, neutral, negative, very negative
  - <https://huggingface.co/datasets/SetFit/sst5>
- The labelled Call Centre Dialog (CCD) dataset
- Synthetic dialog data generated with GPT-4o and labelled with 5 levels of sentiment polarity.

#### Trained Models:

- Logistic Regression
  - o Accuracy: 0.59, Weighted Avg. F1: 0.53
- Multinomial Naive Bayes (MNB)
  - o Accuracy: 0.55, Weighted Avg. F1: 0.41
- distilBERT
  - o Accuracy: 0.68, F1: 0.67, test runtime: 11.16
- tinyBERT
  - o Accuracy: 0.67, F1: 0.67, test runtime: 3.08

#### Trained/Fine-tuned Models

<b>Model</b>	<b>Accuracy</b>	<b>Weighted Ave F1</b>	<b>Test Runtime</b>
Logistic Regression	0.59	0.53	-
Multinomial Naive Bayes (MNB)	0.55	0.41	-
BERT-base	-	-	-
distilBERT	0.68	0.67	11.16
mobleBERT	-	-	-
tinyBERT	0.67	0.67	3.08

For POC, I selected tinyBERT as it is only marginally less accurate than distilBERT but is almost 4 times faster. Due to scarce GPU, BERT models were only trained for 3 epochs and BERT-base and mobileBERT were not trained. Further training may result in better performance.

On further testing, both the fine-tuned tinyBERT and fine-tuned distilBERT were significantly inferior to GPT4o to predict 5 levels of sentiment, so this was used instead.

In future iterations, a better set of training data and more compute power will be used to finetune BERT for better performance.

Jupyter Notebook:

- Sentiment\_Clf\_LR\_and\_MNB.ipynb
- sentiment\_clr\_BERT\_DistilBert.ipynb
- sentiment\_clr\_tinyBERT.ipynb

## 3.4 Components of Automated Topic Sentiment Analysis Solution

### 3.4.1 Data Cleaning and Preparation

Refer to 3.3.2

### 3.4.2 Aspect Topic Module

The full definition of Aspect Topic is found in 3.1.1.

This module (Aspect Topic Module) takes as input the classification results from the *product classifier* and *action classifier* and combines it to create the **Aspect Topic**.

Aspect Topic = customer action (intent) + product/solution

Example



	product	action	aspect_topic
<b>9101</b>	debit_card	renew	renew__debit_card
<b>10712</b>	account	link	link__account
<b>6026</b>	mobile_app	access	access__mobile_app
<b>8558</b>	account	renew	renew__account
<b>13872</b>	debit_card	pay	pay__debit_card

Jupyter notebook

- From\_Unlabelled\_To\_AspectTopic\_01.ipynb

### 3.4.3 Aspect Topic Accuracy Modelling

On receiving data from the Aspect Topic module:

- Details will be logged
- a notification will be sent to
  - developer to examine root causes
  - For human labelling.

When a pre-determined threshold is breached, this module will trigger a Incremental Learning with Partial Fitting of one or both classifiers. This action can also be triggered by the developer.

Jupyter notebook

- To be implemented in future iterations.

### 3.4.4 Topic Modelling Module

In this module, we will use topic modelling techniques to discover hidden topics or themes in the

Topic Modelling Methods

- Non-Negative Matrix Factorization (NMF) or
- BERTopic

NMF is faster, simpler, and works well for basic, interpretable topic modelling tasks, especially when computational resources are limited, and we wish to define the number of topics in advance.

BERTopic excels when contextual understanding is key, and you want a more dynamic, adaptive model that can handle complexity, do not wish to define the number of topics in advance, though it requires more resources and is less interpretable than NMF.

#### 3.4.4.1 Non-Negative Matrix Factorization (NMF)

NMF simplifies complex datasets into meaningful components, revealing underlying patterns and topics. It decomposes a data matrix into two smaller matrices, uncovering insights that inform decision-making.

In the Scikit-learn implementation of Non-negative Matrix Factorization (NMF), the topics generated by the model are not ranked by the count of documents. Instead, NMF produces two matrices:

- W (the document-topic matrix)
  - Each row corresponds to a document and each column represents a topic.
  - The values in this matrix indicate the strength of association between each document and each topic. Higher values suggest that a document is more closely related to a particular topic
- H (the term-topic matrix)
  - This matrix shows how terms are distributed across the topics, with higher values indicating more significant terms for each topic

Although NMF does not inherently rank topics by the number of documents associated with them, we derive such rankings by using the document-topic matrix to calculate the “Count of Documents per Topic”.

- `num_documents_per_topic = np.sum(W > 0, axis=0)`
- `ranked_topics = np.argsort(topic_importance)[::-1]` # Sort in descending order

While a higher number of documents can suggest a topic's prevalence, it does not inherently determine its importance as it does not take into account the content and quality of the documents associated with each topic.

Hence, to calculate the importance of each topic, we also evaluate the Term-Topic Matrix (H) to understand the significance of terms for each topic:

- `term_importance = np.sum(H, axis=1)`
- `ranked_terms = np.argsort(term_importance)[::-1]` # Sort in descending order

Then we combine the insights from both “number of documents per topic” and “term importance” to get a composite normalised score i.e. A topic that is prevalent (high score in W) and also has significant terms (high score in H) could be considered more important.

- `combined_scores = num_documents_per_topic + term_importance`
  - Adjust weights as necessary
- `ranked_combined_topics = np.argsort(combined_scores)[::-1]`

Notes:

- The impact of term importance is too small. For now, I am just using number of documents per topic. In future iterations, will consider other methods or using weights to boost/reduce impact.

### Example

First, NMF is performed to obtain the topics in the table below.

	topic_num	topic_importance	num_of_docs_per_topic	term_importance	features	
	3	3	8013.009280	7942	71.009280	hi, ve, really, bank, frustrated, really frust...
	11	11	7263.652048	7230	33.652048	number, sure, account number, account, 1234567...
	0	0	6997.484693	6948	49.484693	card, credit, credit card, activate card, acti...
	9	9	6302.050841	6258	44.050841	thank, help, thanks, alex, appreciate, ll, gre...
	12	12	5765.356806	5739	26.356806	app, card app, mobile, mobile app, use app, ph...
	1	1	5633.245012	5607	26.245012	money, transfer, account, money account, trans...
	8	8	4954.468352	4932	22.468352	atm, cash, cash atm, withdraw, withdrawal, did...
	17	17	4703.636259	4690	13.636259	balance, account balance, check, check balance...
	4	4	4690.775993	4680	10.775993	virtual, virtual card, disposable, disposable ...
	10	10	4641.431982	4623	18.431982	new, new card, card, order, activate, link new...

Next, a large language model is used to help provide a “topic”, what might have “triggered” the topic and an “explanation” for why it created the topic and trigger.

	topic_num	topic_importance	num_of_docs_per_topic	term_importance	features	topic	trigger	explanation
3	3	8013.009280	7942	71.009280	hi, ve, really, bank, frustrated, really frust...	Credit Card Activation Process	Customers needing assistance to activate new c...	The presence of verbs such as 'activate' and '...
11	11	7263.652048	7230	33.652048	number, sure, account number, account, 1234567...	Bank Transfers	Increase in Digital Transactions	The emphasis on 'transfer' and 'add' suggests ...
0	0	6997.484693	6948	49.484693	card, credit, credit card, activate card, acti...	Incorrect Exchange Rate Application	rate applied	The word list includes multiple mentions of 'r...
9	9	6302.050841	6258	44.050841	thank, help, thanks, alex, appreciate, ll, gre...	Customer Service Issue	Frustrated	The repetition and emphasis of the verb 'frust...
12	12	5765.356806	5739	26.356806	app, card app, mobile, mobile app, use app, ph...	Growing Use of Disposable Virtual Cards	Increase in Ordering of Disposable Virtual Cards	The emphasis on 'order' and 'work' suggests a ...

Jupyter Notebook:

- Topic\_Modeling\_NMF\_v05.ipynb

#### 3.4.4.2 BERTopic

BERTopic helps identify and group similar topics within textual data, simplifying the analysis of customer interactions, feedback, and other text-based communications. BERTopic uses transformer-based embeddings (e.g., BERT) to understand the context and meaning of words, not just their appearance.

Example

	Topic	Count	Name	Representation	Representative Docs
0	0	559	0_thank_thats_thanks_ill	[thank, thats, thanks, ill, alright, thank hel...	[No, that's all. Thank you for your help, No,...
1	1	410	1_bank_current account_current_account bank	[bank, current account, current, account bank,...	[That's perfect, Alex. Thank you so much for y...
2	2	264	2_alex_thank alex_help alex_thanks alex	[alex, thank alex, help alex, thanks alex, tha...	[No, that's all for now. Thank you, Alex, Tha...
3	3	187	3_debit card_debit_new debit_received new	[debit card, debit, new debit, received new, h...	[Hi, I recently received a new debit card from...
4	4	162	4_fixed deposit_fixed_deposit_deposit bank	[fixed deposit, fixed, deposit, deposit bank, ...	[Hi, I'm interested in opening a fixed deposit...

Next, a large language model is used to help provide a “topic” and an “explanation” for why it created the topic.

	Topic	Count	Name	Representation	Representative_Docs	topic	explanation
0	0	559	0_thank_thats_thanks_ill	[thank, thats, thanks, ill, alright, thank hel...	[No, that's all. Thank you for your help., No...	Gratitude and Confirmation	The representative words and documents center ...
1	1	410	1_bank_current account_current_account bank	[bank, current account, current, account bank...	[That's perfect. Alex. Thank you so much for y...	Customer Satisfaction and Account Services in ...	The representative words and documents highlig...
2	2	264	2_alex_thank alex_help alex_thanks alex	[alex, thank alex, help alex, thanks alex, tha...	[No, that's all for now. Thank you, Alex., Tha...	Expressing Gratitude and Acknowledgment to Alex	The representative words and documents suggest...
3	3	187	3_debit card_debit_new debit_received new	[debit card, debit, new debit, received new, h...	[Hi, I recently received a new debit card from...	Activating New Debit Card	The representative words and documents focus o...
4	4	162	4_fixed deposit_fixed_deposit_deposit bank	[fixed deposit, fixed, deposit, deposit bank, ...	[Hi, I'm interested in opening a fixed deposit...	Opening a Fixed Deposit Account	The representative words focus on terms relate...

Jupyter notebook

- Topic\_Modeling\_NMF\_v05.ipynb

## 3.4.5 Sentiment Classification Calculation

### 3.4.5.1 Aspect Topic Sentiment Classification

The 5 Level Sentiment classifier (3.4.5) will classify the sentiment polarity of each dialog text. (below)

```
(1b)
date aspect_topic sentiment dlg_id
==== =====
2024-10-02 | activate__credit_card | Negative | 323
2024-10-02 | activate__credit_card | Negative | 611
2024-10-02 | pay__credit_card | Positive | 298
```

To determine the sentiment of any specific aspect topic, we take the average sentiment of all dialog rows with the same aspect topic.

### 3.4.5.2 Non-Negative Matrix Factorisation (NMF) Sentiment Classification

Example of a NMF topic's top key words:

Topic #1:

- card, new, payment, new card, card payment, use, use card, activate, need, declined

If we were to run a sentiment classifier on top key words (features) of a NMF topic, most, if not all topics will end up with neutral sentiment because most topics consists of “product” words such as “card”, “credit card”, “atm” etc, which are neutral.

Hence, to determine the sentiment polarity of a NMF topic, we run the sentiment classifier on “Trigger” (see 3.4.4.1) instead of the “Features”.

Steps:

- Send prompt to LLM to obtain “most likely topic”, “trigger” and “explanation”.
- Perform Sentiment Classification on column “Trigger”

Feature Names	Most Likely Topic	Trigger	Explanation	Sentiment
card, new, payment, new card, card payment, use, use card, activate, need, declined	New Card Activation Issues	A customer's new payment card was declined during first use	The presence of words like 'new card', 'activate', and 'declined' suggests the issue is related to setting up and using a newly issued payment card.	Negative
...	...	...	...	...

Then we can get a plausible explanation of the topic and an appropriate sentiment polarity.

Methods/Models:

- See 3.3.6.3 – Sentiment Classifier

### 3.4.5.3 BERTopic Sentiment Classification

Similar to 3.4.5.2, if we perform sentiment classification on the “Representation”, it is likely that we will obtain “neutral” most of the time.

Date	Topic	Count	Name	Representation	Representative Docs	Sentiment Polarity
2024-10-02	1	124	1_savings_savings account_account savings_savings savings	['savings', 'savings account', 'account savings', 'savings savings', 'srinivas', 'use savings', 'send', 'check savings', 'transfer savings', 'xiaoxue']	['go savings account', 'how about my savings account', 'go to my savings account please']	neutral
2024-10-02	2	72	2_card payment_payment_payment didnt_payment did	['card payment', 'payment', 'payment didnt', 'payment did', 'work card', 'work', 'didnt work', 'did work', 'payment work', 'payment card']	['my card payment didnt work', 'the card payment i made didnt work', 'i made a card payment but it didnt work why not']	negative

Hence, Sentiment Classification (last column in above image) is performed on the Representative Docs (4<sup>th</sup> Column above).

The example table shows the result of the Sentiment Polarity in the rightmost column.

### 3.4.6 Sentiment Accuracy Monitoring

After deployment, it is inevitable that there will be cases of misclassification of sentiment polarity.

Sampling checks will be done with human-in-the-loop validation. Once a threshold is breached, the Sentiment classifier fine-tune process will be triggered. This action can also be triggered by the developer.

If the number of misclassifications is not huge, we can manually label all of it and do a normal fine-tune of the model. However, if the number of misclassifications is large, this method may not be cost-efficient.

Note:

Misclassification may be identified via user (or large numbers of users) flagging so we know there is an issue but there may be no label.

Unlike the case of binary “spam” vs “not spam”, it is more difficult for users to label 5 levels of sentiment polarity consistently, so we might not want users to do the labelling.

#### Jupyter notebook

- To be implemented in future iterations.

### 3.4.7 Topics Database

The Topics Database will store the results of Aspect Based Sentiment Mining and Topic Modelling.

Example data stored:

Aspect Based Sentiment Mining:

```
(1b)
date aspect_topic sentiment dlg_id
==== =====
2024-10-02 | activate__credit_card | Negative | 323
2024-10-02 | activate__credit_card | Negative | 611
2024-10-02 | pay__credit_card | Positive | 298
```

Topic Modelling

```
(2)
date      topic  count  top_terms_in_each_topic
=====
2024-10-02 | -1    362   -1_phone_work
2024-10-02 | 0     78    0_savings_rate_good
2024-10-02 | 1     52    1_loans_high_fast
2024-10-02 | 2     47    2_mobile_app_speed
```

The Customer Satisfaction Metrics (3.4.8) module will use these data to do their calculations to display in the dashboard.

#### Jupyter notebook

- Not Applicable

### 3.4.8 Customer Satisfaction Metrics

#### *3.4.8.1 Net Promoter Score (NPS)*

NPS calculation is applied to Aspect Based Sentiment Analysis but not for Topic Modelling.

As the sentiment analysis has 5 levels, we can assign a number range to it in this manner:

Sentiment	Very Negative	Negative	Neutral	Positive	Very Positive
NPS Score	0	2.5	5	7.5	10

Based on their ratings, categorize respondents into three groups:

- Promoters (9-10):
  - Loyal enthusiasts who will continue to buy and refer others.
- Passives (7-8):
  - Satisfied but unenthusiastic customers who are vulnerable to competitive offerings.
- Detractors (0-6):
  - Unhappy customers who can damage your brand through negative word-of-mouth.

To determine the percentage of respondents in each category:

$$\begin{aligned}
 - \text{Percentage of Promoters} &= \frac{\text{Number of Promoters}}{\text{Total Respondents}} \times 100 \\
 - \text{Percentage of Detractors} &= \frac{\text{Number of Detractors}}{\text{Total Respondents}} \times 100
 \end{aligned}$$

To calculate NPS

$$- \text{NPS} = \text{Percentage of Promoters} - \text{Percentage of Detractors}$$

The resulting score can range from -100 to +100.

A positive score indicates more promoters than detractors, while a negative score indicates the opposite. An NPS above 0 is considered good, above 20 is favourable, above 50 is excellent, and above 80 is world-class.

[6]:

	aspect_topic	avg_sentiment_score	nps_group	rep_dlg_id	rep_text
92	update_atm	6.309524	detractors	2024-10-07_1017--3530	if i want to change my pin number can i direct...
348	apply_website	4.021739	detractors	2024-10-14_dlg_1230	I think applying online would be convenient fo...
211	open_forex_account	5.312500	detractors	2024-10-14_dlg_1100	I want to open a forex account that has compet...
235	access_account	5.113636	detractors	2024-10-07_1017--2859	i have an emergency and lost my phone i need ...
374	unlock_website	3.500000	detractors	2024-10-14_dlg_1906	This is ridiculous. The last transaction shoul...

[7]:

	total_respondents	num_promoters	pct_promoters	num_detractors	pct_detractors	nps
0	393	5	1.272265	352	89.56743	-88.295165

[Jupyter notebook](#)

- [nps\\_calculations.ipynb](#)

#### 3.4.8.2 Customer Satisfaction Score (CSAT)

CSAT calculation is applied to Aspect Based Sentiment Analysis but not Topic Modelling.

From the 5-level sentiment polarity, we transform it to the following scores:

Sentiment	Very Negative	Negative	Neutral	Positive	Very Positive
CSAT Label	Very Unsatisfied	Unsatisfied	Neutral	Satisfied	Very Satisfied
CSAT Score	1	2	3	4	5



To calculate the CSAT Score:

$$CSAT\ Score = \left( \frac{Number\ of\ Satisfied\ Responses}{Total\ Responses} \right) \times 100$$

where number of satisfied responses includes "Satisfied" and "Very Satisfied".

For example, if 100 bank customers (or 100 customer dialog as proxy) were surveyed and we received the following results:

- Satisfied Responses (scores of 4 and 5) = 70
- Total Responses = 100

Then the CSAT score would be 70/100 i.e. 70%. This means that 70% of respondents were satisfied with their experience with the bank.

[7]:

	aspect_topic	avg_csat_lbl_score	csat_group	rep_dlg_id	rep_text
266	reset__mobile_app	2.970588	neutral	2024-10-07_1017--844	can i use app to reset pin attempts
227	unblock__loan	3.066667	neutral	2024-10-14__dlg_495	It was blocked just yesterday when I was tryin...
222	dispute__card	3.083019	neutral	2024-10-07_1017--409	the exchange rate from my card payment isnt right
65	deposit__card	3.601449	satisfied	2024-10-07_1017--770	can i add money to my card automatically durin...
12	select__website	4.166667	satisfied	2024-10-14__dlg_160	That sounds exactly like what I need! I really...
299	activate__debit_card	2.847826	unsatisfied	2024-10-14__dlg_171	Yes, I understand. I've considered those point...
232	update__debit_card	3.055556	neutral	2024-10-14__dlg_1145	That sounds perfect, Alex. I really appreciate...
325	refund__loan	2.750000	unsatisfied	2024-10-14__dlg_1980	Sure, the loan account number is 123456789. I ...
238	block__debit_card	3.037736	neutral	2024-10-07_1017--6872	freeze my debit card i want to stop at all tra...
373	report__loan	2.400000	unsatisfied	2024-10-14__dlg_1051	I shouldn't have to deal with this stress! The...

	num_satisfied	total_responses	csat_score
0	218	393	55.470738

[Jupyter notebook](#)

- csat\_calculations.ipynb



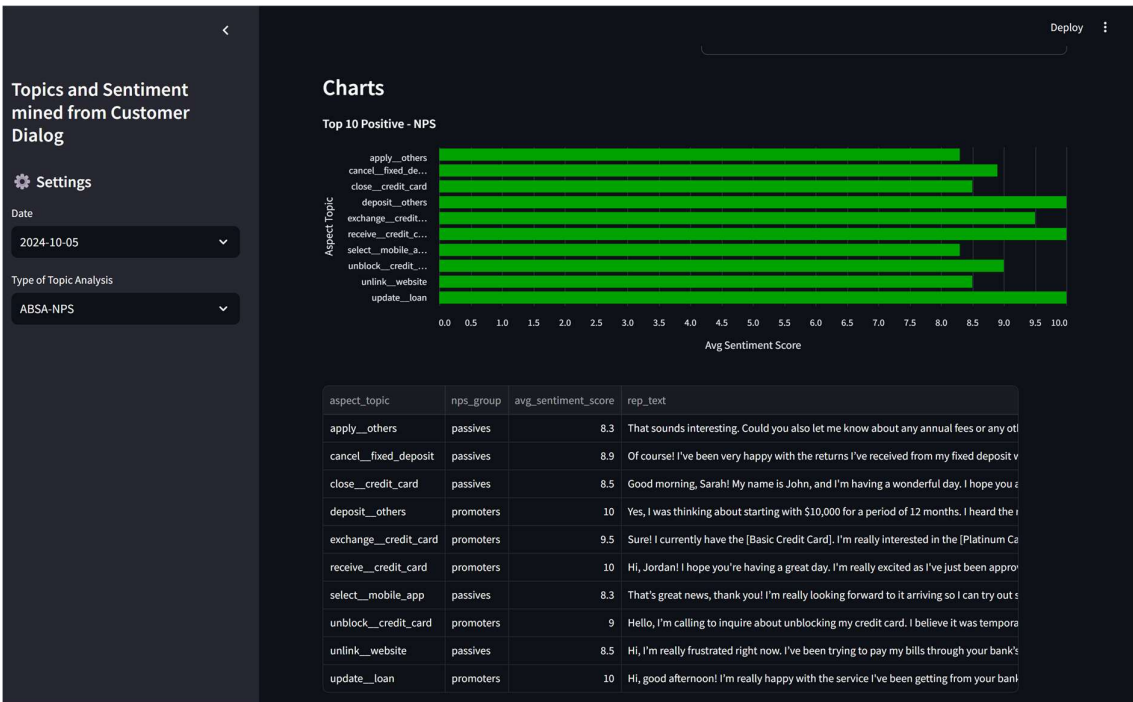
### 3.4.9 Dashboard

The dashboard is created using Streamlit.

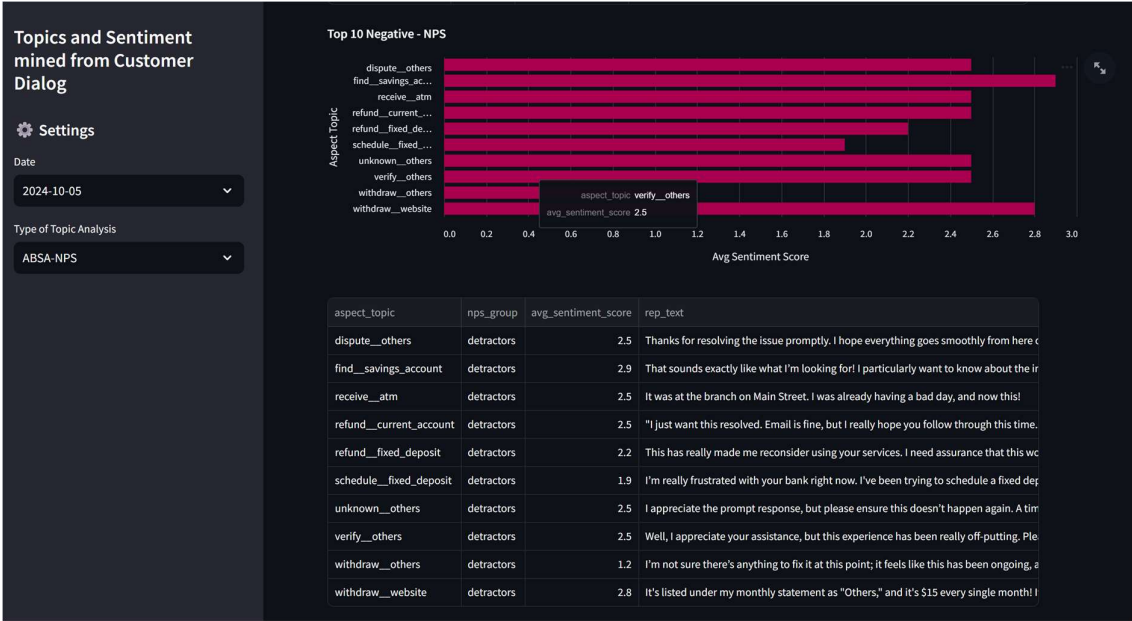


Left side bar: Select *Date* and Select *Type* of Topic Analysis

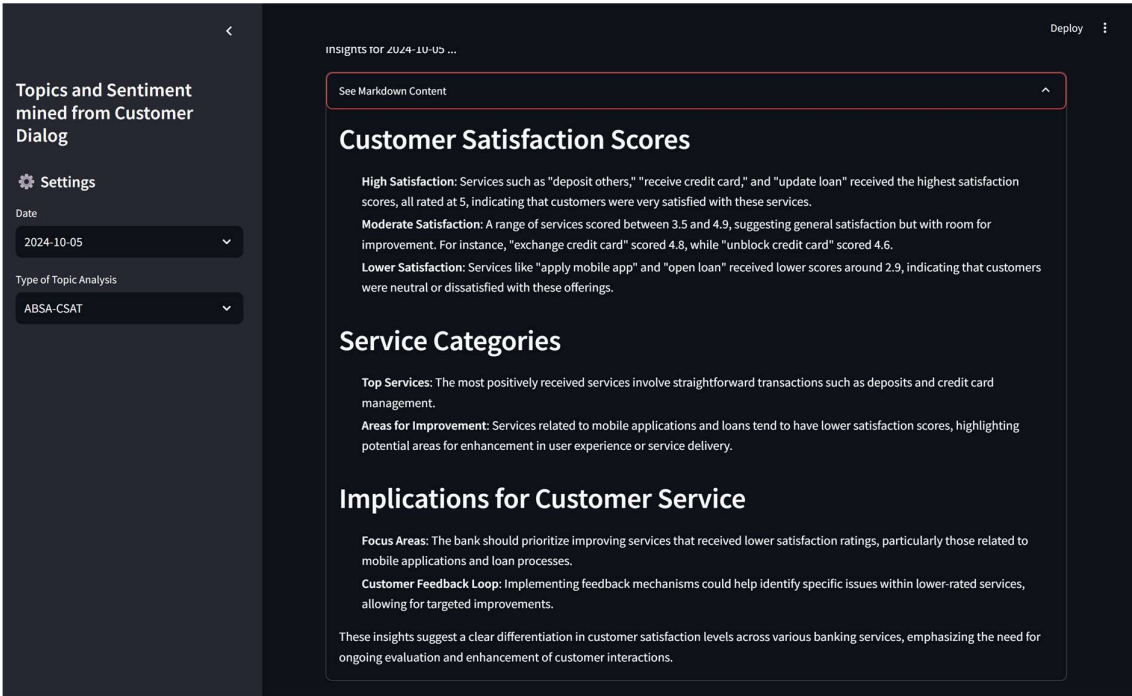
Right: As Aspect Topic Sentiment Analysis selected, the Net Promoter Score is shown as well as distribution of Detractors, Passives and Promoters.



The Top 10 Positive topics are displayed in a bar chart, followed by a table with details.



The Top 10 Negative topics are displayed in a bar chart, followed by a table with details.



The Insights and Recommendations in the screen shot above was generated by LLM from the topic analysis data. However, it is not yet integrated into the application code. The example above is hard-coded and only appears for ABSA-CSAT to serve as an example and a to-do for future iterations.

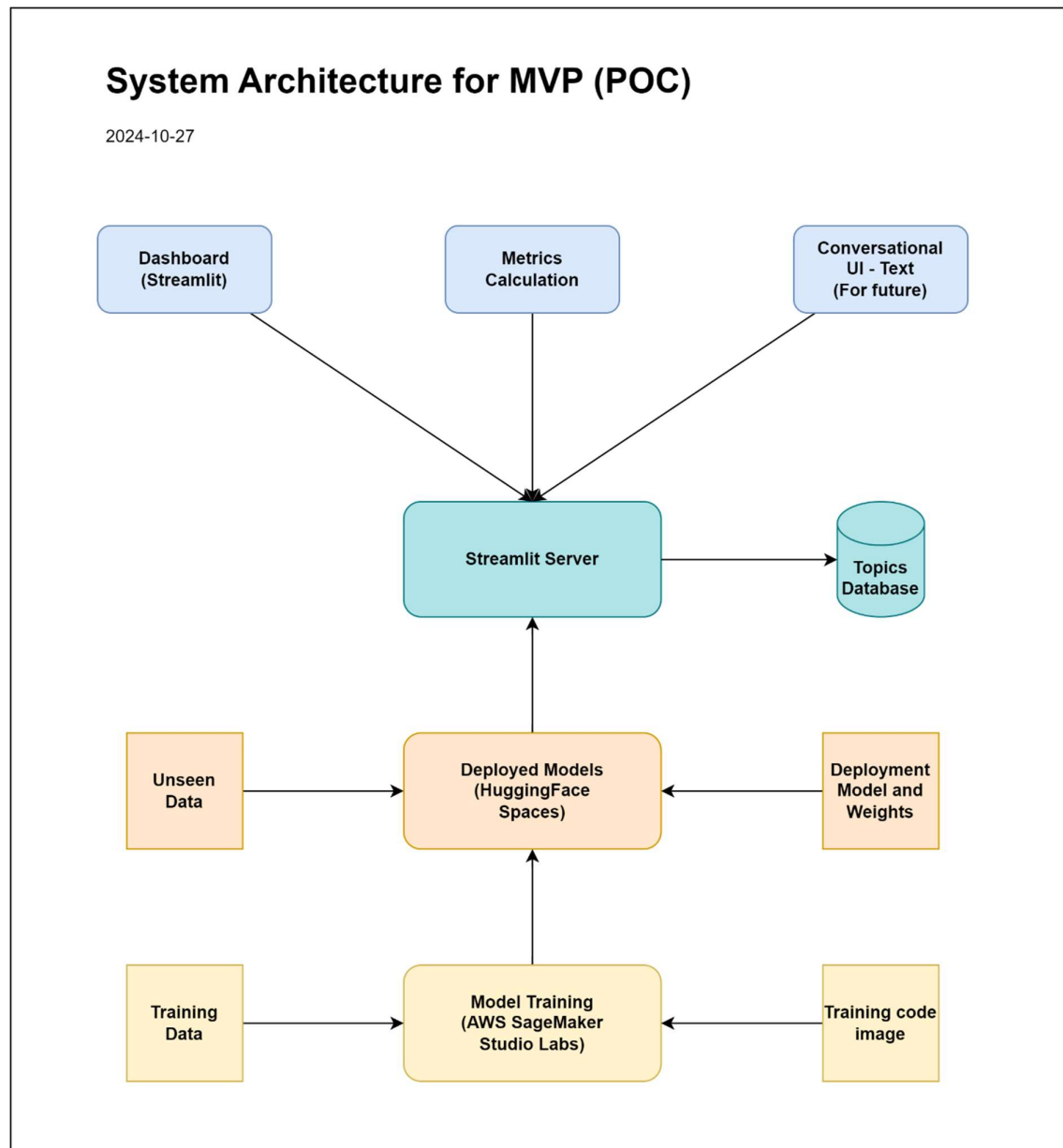
### 3.4.10 Conversation User Interface

Not implemented in this iteration.

Jupyter notebook

- Not applicable.

## 4.0 System Architecture



Data Ingestion, data preparation and model training are done locally if GPU is not required e.g. Logistic Regression, MNB.

Product, Action and Sentiment classifiers using BERT transformer architecture is extremely slow if fine-tuning using CPU and these were trained on AWS SageMaker Studio Labs which has GPU.

After all machine learning models have been trained or fine-tuned, they can available in HuggingFace hub and can be downloaded to run inference locally or hosted on HuggingFace Spaces to perform inference using cloud computing.

Metrics calculation is done on-demand and pulls data from the Topics database. The results are cached for faster responses.

The dashboard (3.4.9) is built using python Streamlit.

If time permits, a conversational text user interface will be incorporated in the dashboard.

## 5.0 Conclusion / Area of Improvement

- Relevant banking customer dialog dataset are few. Need to find more to cover more products, actions and sentiment (preferable 5 levels) and avoid class imbalance.
- Due to time constraints and computer power issue, Aspect Topic is a small subset of all possible bank products and services, and related action intent. This should be expanded in future together with obtaining more suitable high quality dataset.
- Dataset do not have timestamp, so we cannot show topic trend changes across time. For this POC, I have generated a second dataset from LLM, but it takes hours (and money) to generate just a small dataset. With just 2 time periods, a trend analysis is not meaningful. But if this system is used in a real deployment, the issue of not having data over different time periods would probably be resolved easily.
- Embedded chatbot interface within the dashboard is not implemented due to lack of time. It would be good to have a chatbot interface to turn natural language change the contents displayed in the dashboard in future.

## 6.0 Bibliography & Appendix

### 6.1 Appendix A - Project Files

- Project Report
  - <https://github.com/atsui888/Practical-Language-Processing/tree/main/Project-Report>
- Video
  - <https://github.com/atsui888/Practical-Language-Processing/tree/main/Project-Report>
- Code
  - <https://github.com/atsui888/Practical-Language-Processing/tree/main/Code>

### 6.2 Appendix B – Datasets Used

#### Banking77

- This dataset is composed of online banking queries annotated with their corresponding intents.
- Provides a very fine-grained set of intents in a banking domain.
- It comprises 13,083 customer service queries labelled with 77 intents.
- <https://huggingface.co/datasets/PolyAI/banking77>

#### Minds14

- MINDS-14 is training and evaluation resource for intent detection task with spoken data.
- It covers 14 intents extracted from a commercial system in the e-banking domain, associated with spoken examples in 14 diverse language varieties.
- <https://huggingface.co/datasets/PolyAI/minds14>

#### FiNER\_139

- FiNER-139 is comprised of 1.1M sentences annotated with XBRL tags extracted from annual and quarterly reports of publicly traded companies in the US.
- Has label set of 139 entity types.
- Another important difference from typical entity extraction is that FiNER focuses on numeric tokens, with the correct tag depending mostly on context, not the token itself.
- <https://huggingface.co/datasets/nlpauieb/finer-139>

#### Stanford Sentiment Treebank

- 5 labels: very positive, positive, neutral, negative, very negative
- Training data is on sentence level, not on phrase level
- <https://huggingface.co/datasets/SetFit/sst5>

#### Schema-Guided Dialog

- The Schema-Guided Dialogue dataset (SGD) was created for the Dialogue State Tracking Challenge (DSTC8), focusing on multi-domain, task-oriented conversations.

- It includes over 18,000 annotated dialogues across 17 domains, such as banking, travel, and weather, designed to evaluate various tasks like intent prediction and state tracking
- [https://huggingface.co/datasets/google-research-datasets/schema\\_guided\\_dstc8](https://huggingface.co/datasets/google-research-datasets/schema_guided_dstc8)

#### Synthetic Generated Dialog Dataset

- As the Schema-Guided Dialog dataset is not a large dataset and is already used for training and validation, synthetic dialog data with varying sentiment polarity will be generated using multiple Large Language Models for diversity to act as the unseen test data.

## 6.3 Appendix C - Citations

#### *BERTopic*

- <https://github.com/MaartenGr/BERTopic>
- <https://towardsdatascience.com/bertopic-what-is-so-special-about-v0-16-64d5eb3783d9>

#### *Net Promoter Score*

- <https://www.surveymonkey.com/mp/nps-calculator/>
- <https://blog.hubspot.com/service/how-to-calculate-nps>
- <https://userpilot.com/blog/nps-dashboard/>
- <https://userguiding.com/blog/nps-dashboard>
- <https://public.tableau.com/app/profile/mirjam.haring/viz/NetPromoterScoreDashboard/NPSDashboard>

#### *Non-Negative Factorisation (NMF)*

- <https://stackoverflow.com/questions/51676677/how-to-get-frequencies-of-topics-of-nmf-in-sklearn>
- <https://www.freecodecamp.org/news/advanced-topic-modeling-how-to-use-svd-nmf-in-python/>
- [https://scikit-learn.org/stable/auto\\_examples/applications/plot\\_topics\\_extraction\\_with\\_nmf\\_lda.html](https://scikit-learn.org/stable/auto_examples/applications/plot_topics_extraction_with_nmf_lda.html)

#### *Customer Satisfaction Score*

- <https://www.surveysensum.com/blog/how-to-calculate-csat-score>
- <https://www.callcentrehelper.com/how-to-calculate-customer-satisfaction-csat-109557.htm>

#### *HuggingFace*

- (Datasets) <https://huggingface.co/docs/datasets/index>

#### *Ticket Resolution Rate*

- <https://www.metrichq.org/support/ticket-resolution-rate/>
- <https://www.gorgias.com/blog/first-contact-resolution>
- <https://www.zendesk.com/sg/blog/first-contact-resolution-friend-foe-frenemy/>
- <https://www.geckoboard.com/best-practice/kpi-examples/average-resolution-time/>

## 6.4 Appendix D: Usage of ChatGPT

I use ChatGPT sometimes as a substitute for a search engine, however, the usage was not a lot. Currently, for search information related to coding, I still prefer going to Stack Overflow directly because the solution and the accompanying explanation is usually peer reviewed.

In contrast, in my experience, code suggestion from ChatGPT, while looking impressive, frequently have issues with running in actual computing environment and is more suitable for simpler code scenario or boiler plate code e.g. almost everyone's Sklearn training loop looks similar because Sklearn has made the code very high level of abstraction. Same situation for HuggingFace Pipeline Class used in inferencing, it is highly abstracted and almost everyone's code will look the same.

As this Dialog Topic Analysis System is using many techniques, and outputs not just topic with importance, but also metric scoring like NPS and CSAT in one complex system, it will be too painful to try generating this system code using LLM and then to troubleshoot it with natural language command.

But maybe someday in the future, this could be done, but not today ... then I will spend more time doing solution architecting rather than writing so much code.

What GPT-4o was used for is the below, via API calls:

- Creating of Synthetic Multi-turn Dialog data
  - o Very useful! It would be impossible for me to generate 2,000 multi-turn dialog on banking domain manually.
- Sentiment Classification:
  - o Although I trained logistic regression, MNB and also finetuned BERT-base, distilBERT, mobileBERT and tinyBERT, the performance was not good.
  - o I think this is because my assembled training dataset from many sources has a class imbalance and is overall too small.
  - o Hence, after training, fine-tuning transformer and testing, I decided to use GPT-4o for sentiment classification (5 levels), otherwise, the results and subsequent analysis in the dashboard would be significantly adversely affected.
  - o However, for Product Classification and Action Classification, GPT-4o is not used, instead, the fine-tuned BERT based transformers have good performance and are used in this project.