Customer Support Bot Analysis Report

1. Data Loading and Initial Exploration

The dataset consists of **678 customer support chatbot conversations** stored in a JSON file. Each entry contains three fields:

- `index`: Unique identifier for each conversation.
- `conversation`: The full text of the conversation, including both user and agent messages.
- `category`: The preliminary categorization of the conversation (e.g., "Order Creation").

Dataset overview:

- Shape: `(678, 3)` — 678 conversations with 3 columns.
- Data types: `index` (int64), `conversation` (object), `category` (object).
- Missing values: Only 2 conversations had missing text.

A sample conversation illustrates the structure, where messages from the user and agent are concatenated using `:::` as a delimiter:

```
User: مساء الخير
:::Agent: مرحبا بك في بوسطة! 👋 أنا بسام، المساعد الذكي...
:::User: هل متاح استلام جزئي
:::Agent: للأسف، بوسطة مش بتوفر استلام جزئي...
```

Initial observations:

- The `conversation` column contains multi-turn dialogues.
- Conversations are predominantly in Arabic, and include both user queries and automated agent responses.
- Some conversations are incomplete or automatically terminated due to inactivity.

This initial exploration confirms the dataset is ready for further **conversation analysis, clustering, and handoff detection**.

2. Data Cleaning and Preprocessing

To prepare the conversations for analysis, a preprocessing step was performed to clean the text data. The main goals were to:

1. Remove special separators (`:::`) used in the raw conversation logs.
2. Strip speaker labels such as `User:` and `Agent:` to simplify the text.
3. Remove extra whitespace for consistency

Observations:

- The cleaned text now contains only the message content without labels or separators.
- This preprocessing ensures the conversations are uniform and ready for **clustering, sentiment analysis, and handoff detection**.

3. Conversation Clustering

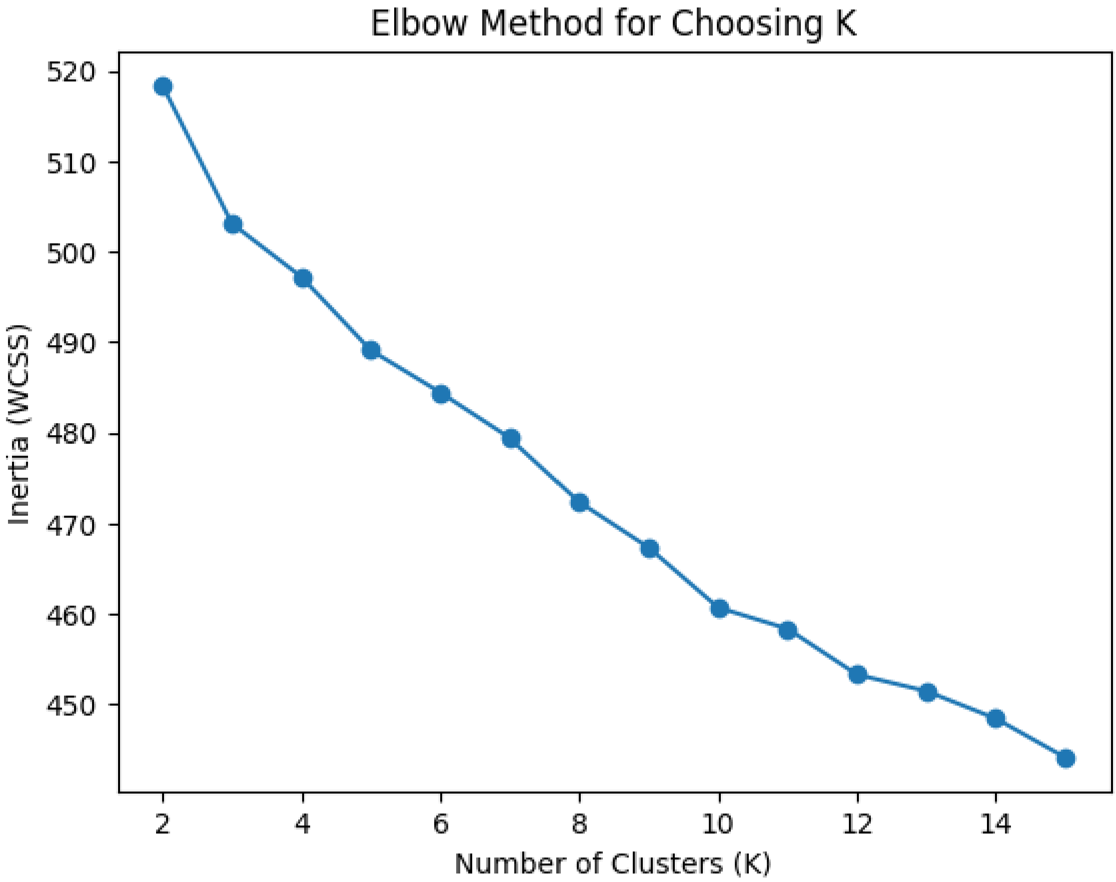
To group conversations by their main topic, we applied **TF-IDF vectorization** followed by **K-Means clustering**. The process included the following steps:

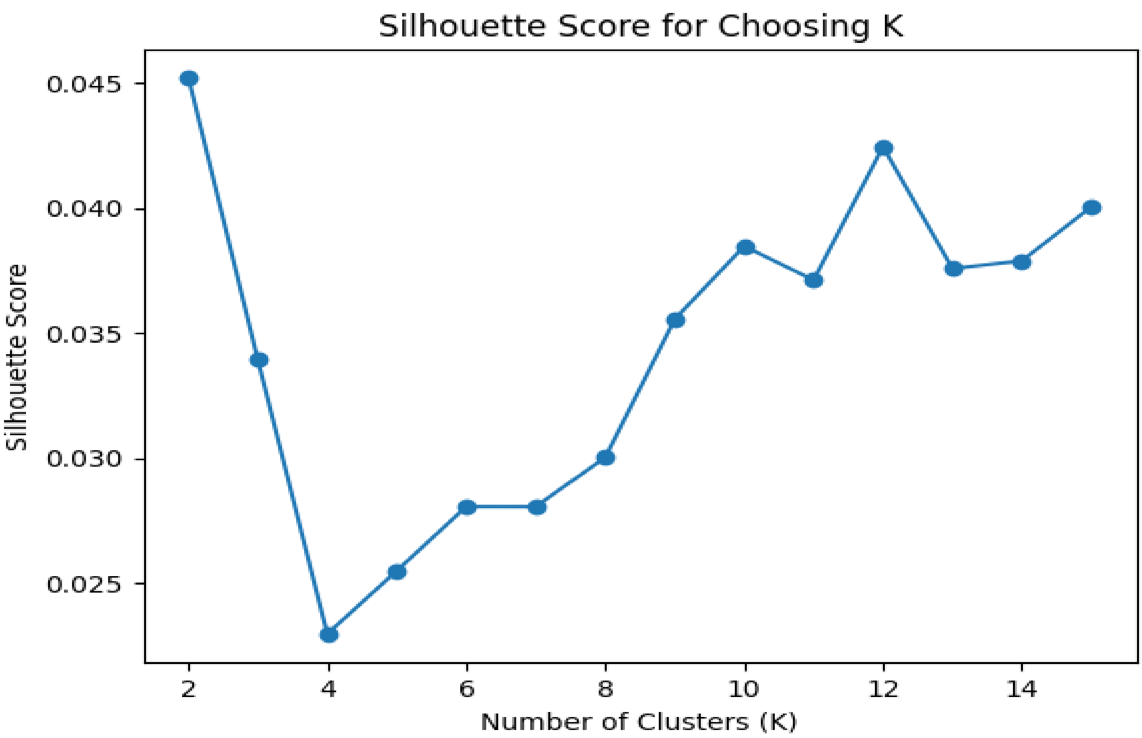
3.1 Feature Extraction

- TF-IDF (Term Frequency–Inverse Document Frequency) was used to convert the cleaned conversations into numerical feature vectors.
- Both **unigrams and bigrams** were included to capture key phrases.
- Arabic stopwords and custom common terms (like greetings and polite words) were removed to reduce noise and improve clustering quality.

3.2 Determining Optimal Number of Clusters

- The **Elbow Method** was used to analyze within-cluster sum of squares (WCSS).
- **Silhouette scores** were also calculated to validate cluster separation.
- Based on these metrics, **12 clusters** were chosen for K-Means.


Elbow Method for Choosing K
Inertia (WCSS)
520
510
500
490
480
470
460
450
2
4
6
8
10
12
14
Number of Clusters (K)


Silhouette Score for Choosing K
Silhouette Score
0.045
0.040
0.035
0.030
0.025
2
4
6
8
10
12
14
Number of Clusters (K)

3.3 K-Means Clustering

- K-Means was applied to the TF-IDF feature matrix.
- Each conversation was assigned a cluster label.
- Top terms per cluster were extracted to interpret each cluster's main topic.

3.4 Cluster Interpretation

Based on the top terms in each cluster, the clusters were labeled as follows:

Cluster	Count	Suggested Label	Type
0	18	Packaging Materials Inquiry	Customer Intent
1	91	Bot Idle / Connection Check	Bot Template
2	124	App Navigation / How-To	Customer Intent
3	64	Closing / Feedback Request	Bot Template
4	50	Delivery Issue / Courier Problem	Customer Intent
5	51	Shipment Tracking	Customer Intent
6	54	Resolution Message / Courier Follow-up	Mixed
7	31	Pricing + Hotline Inquiry	Customer Intent
8	27	General Complaint / Unclear	Noisy Intent
9	35	Escalation / Support Handling	Support Action
10	73	Delivery Complaint	Customer Intent
11	60	Escalation to Specialist (Handoff)	Support Action

Observations:

- Clusters 1 and 3 represent automated bot templates for idle connections and closing messages.
- Clusters 0, 2, 4, 5, 7, and 10 capture **customer intents** such as delivery inquiries, tracking, pricing, and packaging.
- Clusters 9 and 11 correspond to **support actions**, indicating cases where escalation or handoff to a human agent occurred.
- Some clusters (e.g., 8) represent noisy or unclear conversations.

Conclusion:
Clustering successfully grouped the conversations by their main reason for contact, enabling a clear distinction between **customer intents**, **bot-generated messages**, and **support actions** including handoffs.

4. Handoff Detection (Bot → Manual Agent)

To identify conversations where the bot handed over the user to a human agent, a **keyword-based detection method** was implemented.

- Keywords indicating handoff included phrases such as: "معاك","من خدمة العملاء","تم تحويلك","ممثلة خدمة العملاء".
- If any of these keywords were present in the cleaned conversation text, the conversation was marked as a **handoff**.

Handoff Detected	Count
Yes (1)	487
No (0)	191

Observation:
A majority of conversations (~72%) involved escalation or intervention by a human agent, highlighting that handoff occurs frequently for customer issues that the bot cannot fully resolve.

5. Sentiment Analysis

To assess customer sentiment during the conversations, a **lexicon-based approach** was used:

- **Positive indicators:** "شكرا","تمام","كويس","ممتاز","حلو".
- **Negative indicators:** "مشكلة","مش شغال","زعلان","سيء","مش راضي","مستعجل","ضروري".

Sentiment	Count
Positive	357
Neutral	242
Negative	79

Observation:

- Most conversations were **positive** (~53%), suggesting that the bot and support process generally handled customer issues satisfactorily.
- A smaller fraction (~12%) were **negative**, indicating unresolved complaints or urgent issues.

Conclusion:
The combination of **handoff detection** and **sentiment analysis** provides valuable insights into both **customer satisfaction** and **situations requiring human intervention**. This information can guide improvements in bot responses and escalation protocols.

6. Urgency Detection

To prioritize customer requests, a **keyword-based urgency detection** approach was implemented.

- **High urgency keywords:** "ضروري", "حالاً", "دلوقتي", "مستعجل", "مش عارف", "مهم".
- **Medium urgency keywords:** "عايز", "محتاج", "مش فاهم", "ممكن".
- Conversations not containing these keywords were labeled as **Low urgency**.

Urgency	Count
Medium	471
High	153
Low	54

Observation:

- Most conversations (~70%) were of **medium urgency**, reflecting standard customer inquiries.
- About 23% were **high urgency**, indicating issues that may require immediate attention.
- A small fraction were **low urgency**, suggesting routine questions or low-priority requests.

7. Quality of Issue Handling

To assess how effectively issues were addressed, a **handling quality metric** was defined:

- **Resolved:** Signals such as "شكرا", "تمام", "لا شكرا", "متشكر" indicated successful resolution.
- **Unresolved:** Signals like "مش عارف", "مشكلة", "لسه", "مش شغال", "مش فاهم" indicated that the issue was not resolved.
- **Unknown:** Cases with no clear signals.

Handling Quality Distribution:

Handling Quality	Count
Unresolved	336
Resolved	179
Unknown	163

Observation:

- A significant portion (~50%) of conversations remained **unresolved**, highlighting opportunities for improving the bot or support workflow.
- About 26% of conversations were clearly **resolved**, reflecting successful bot or agent assistance.
- ~24% were **unknown**, where the resolution status could not be inferred from the conversation.

Conclusion:
Combining **urgency detection** with **handling quality assessment** enables better prioritization of support tickets and identification of areas where bot responses or human intervention need improvement.

8. Data Visualization

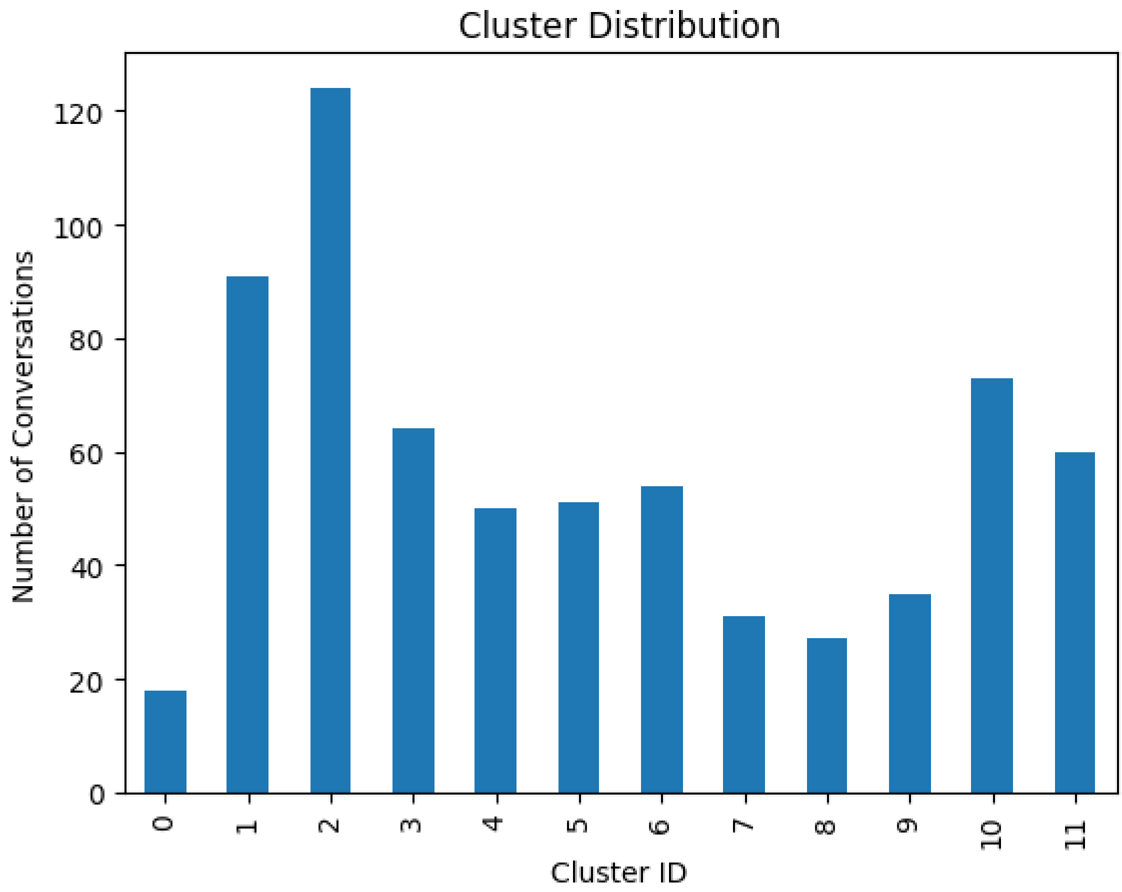
To better understand the distribution of clusters, sentiment, and handoff occurrences, several visualizations were created.

8.1 Cluster Distribution

The bar chart below shows the number of conversations in each cluster.

- Clusters corresponding to customer intents (e.g., delivery issues, shipment tracking) have the largest volumes.

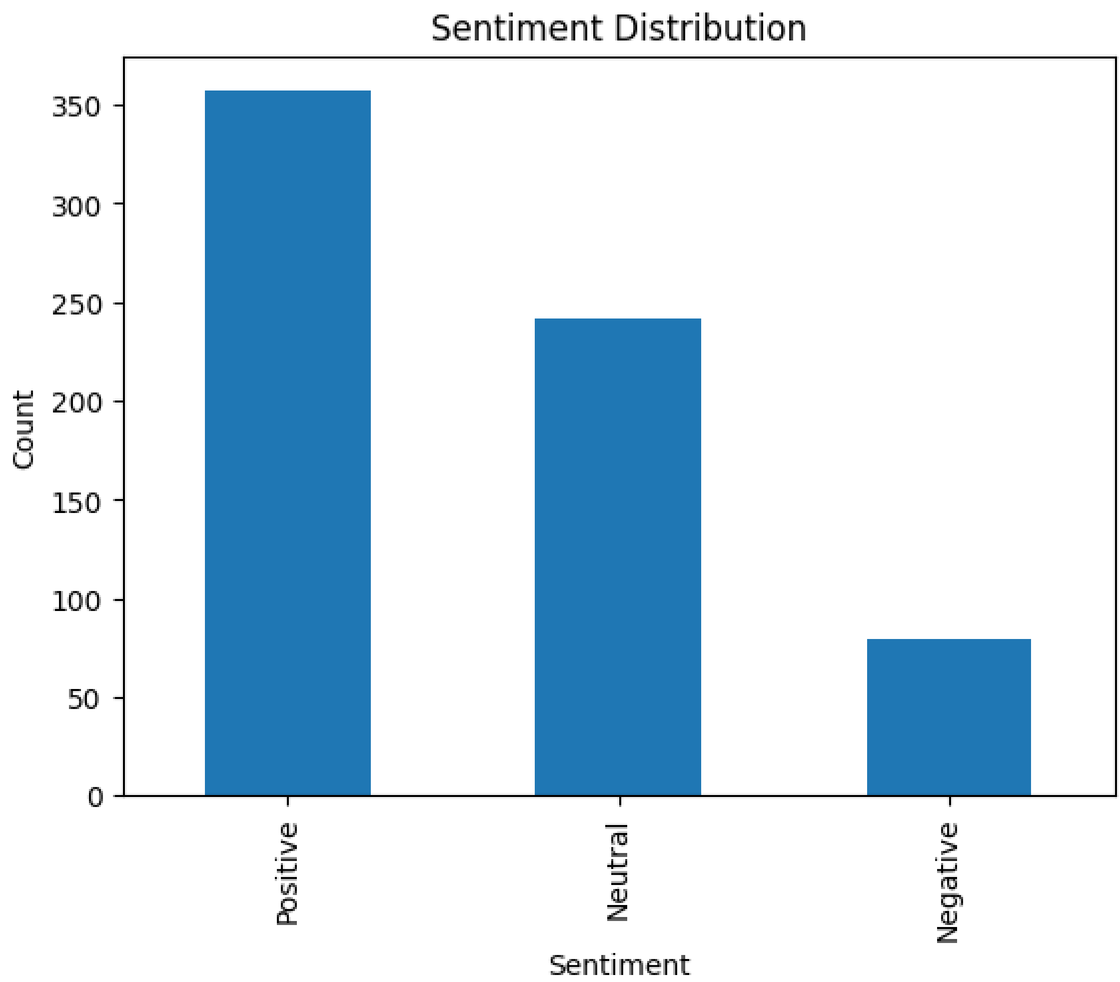
- Bot template clusters (idle checks, closing messages) and less frequent intents are smaller.



8.2 Sentiment Distribution

The sentiment distribution bar chart illustrates customer satisfaction across conversations:

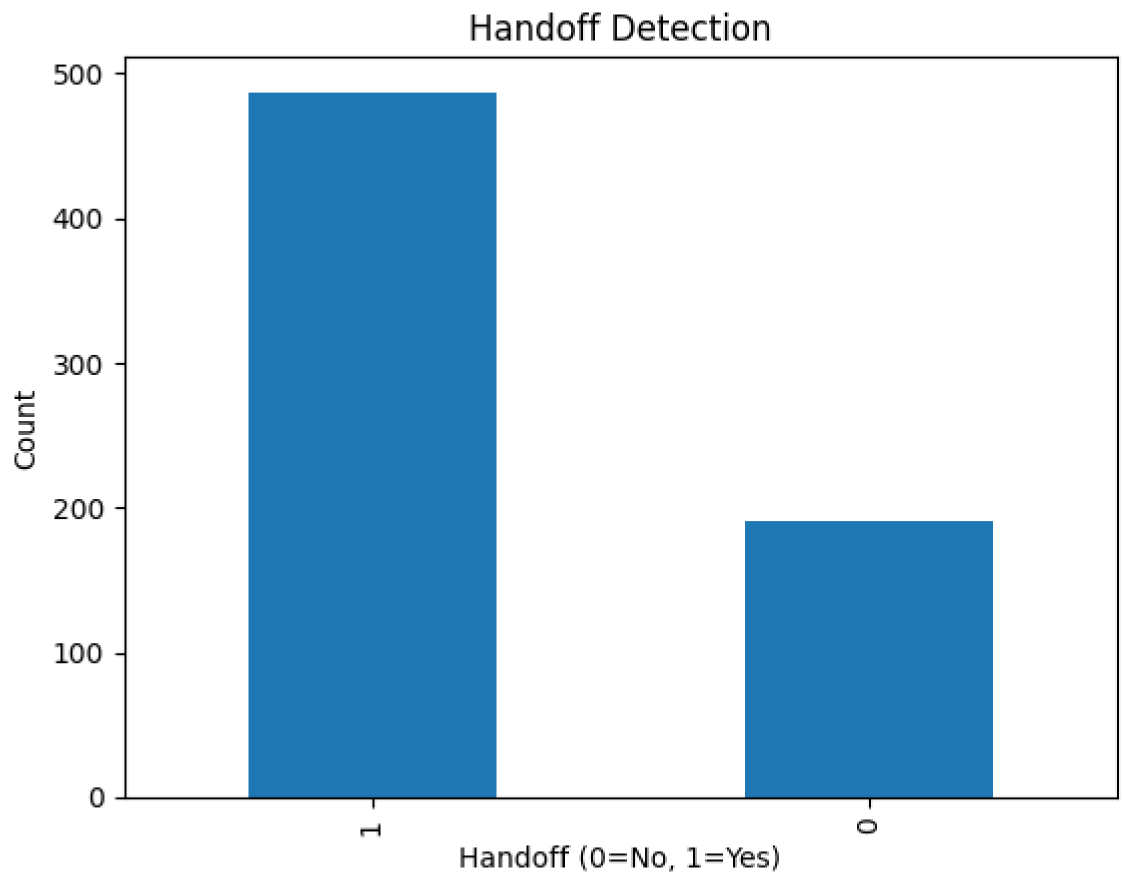
- Positive sentiment is dominant, indicating mostly satisfactory interactions.
- Neutral and negative sentiments are less frequent but highlight unresolved or problematic cases.



8.3 Handoff Detection

The handoff detection bar chart shows the proportion of conversations escalated from the bot to a human agent:

- A majority (~72%) of conversations involved handoff to a human agent.
- This visualization highlights the importance of human intervention for certain customer issues.



Observation:
Visualizations complement the analytical results by providing clear insights into conversation patterns, sentiment trends, and bot escalation rates. They help identify high-volume customer issues and areas requiring process improvements.

9. Agentic AI Design (Conceptual)

To improve efficiency and customer satisfaction, a **next-generation agentic AI system** is proposed. This system combines **NLP, intent classification, sentiment and urgency detection, automated resolution, and human escalation** into a unified support framework.

9.1 High-Level Architecture

Flow Overview:

1. **User Query Input**:
    a. The customer sends a query through the chat interface.
2. **Bot Processing**:

   a. The bot receives the query and passes it to the **NLP Module**.
   b. **Intent Classification**: The NLP module classifies the conversation into a cluster representing the customer's intent (e.g., delivery issue, shipment tracking, pricing inquiry).
   c. **Sentiment & Urgency Detection**: The system evaluates the emotional tone and urgency level of the query.
3. **Automated Response**:
   a. For standard requests, the bot attempts to resolve the issue using a **knowledge base, FAQ, or pre-defined templates**.
4. **Escalation**:
   a. If the query is **high-urgency**, negative sentiment, or cannot be resolved automatically, the conversation is **escalated to a human agent**.
5. **Logging and Learning**:
   a. All conversations are logged for quality monitoring.
   b. Feedback and resolution outcomes are used to **update the knowledge base** and improve the AI model over time.

9.2 Conceptual Flowchart

```
User --> Chatbot --> [NLP Module] --> Intent Detection --> Auto-Response

                |

                v

          Urgency/ Sentiment

                |

                v

           Escalation? --> Human Agent

                |

                v

           Feedback Logging --> Knowledge Base Update
```

9.3 Inputs, Outputs, and Interaction

Component	Input	Output	Interaction
NLP Module	Cleaned user query	Intent label, sentiment, urgency	Provides structured data for decision-making
Automated Response Engine	Intent label	Suggested answer	Pulls from FAQ/templates for instant response
Escalation Module	Unresolved / High urgency query	Human agent handoff	Ensures urgent or complex issues receive manual attention
Logging & Learning	Conversation + Resolution	Updated knowledge base	Continuously improves AI accuracy and response quality

9.4 Benefits of the Agentic AI System

- **Efficiency**: Reduces workload on human agents by resolving routine queries automatically.
- **Prioritization**: Urgency detection ensures critical issues are escalated quickly.
- **Customer Satisfaction**: Sentiment monitoring and proactive escalation improve experience.
- **Continuous Learning**: Logging and feedback loops allow the AI to improve over time, adapting to new queries and patterns.

Conclusion:
This design integrates **intelligent automation with human oversight**, enabling faster, more accurate support while reducing operational overhead and improving overall service quality.