Kura

Making sense of customer data at scale

Converting unstructured data to valuable insights

The Challenge of Conversation Data

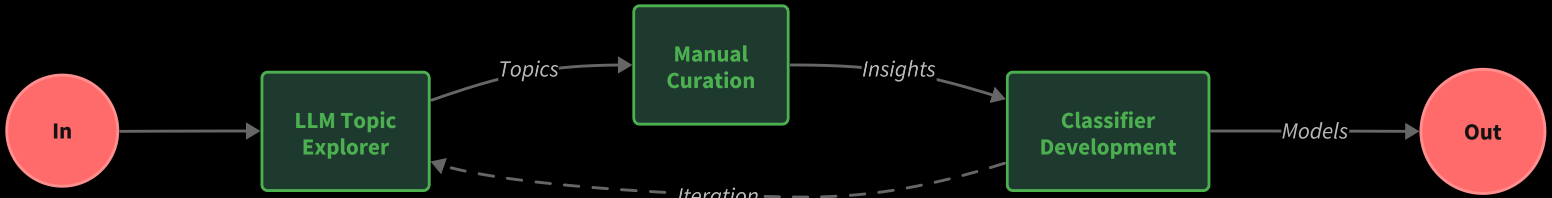
What is Kura?

Getting Started with Kura

```python
from kura import Kura
from kura.types import Conversation
import asyncio

# Initialise the Kura object
kura = Kura()
```

The Core Approach



Loading Different Datasets in Kura

```python
from kura import Kura
from kura.types import Conversation
import asyncio

# Initialise the Kura object
kura = Kura()

# Load in a Conversation from Hugging Face
conversations = Conversation.from_hf_dataset(
    "ivanleomk/synthetic-gemini-conversations", split="train"
)

# Kick off the clustering step
asyncio.run(kura.cluster_conversations(conversations))
kura.visualise_clusters()
```

Beyond Topic Modeling

Feature Prioritization At Anthropic

Summarizing User Conversations

```python
from pydantic import BaseModel

# Define our Response Model
class UsagePattern(BaseModel):
    category: Literal["Learning", "QuestionAnswering", "Brainstorming", "TaskCompletion"] = Field(
        description="The primary usage pattern of this conversation"
    )
    summary: str = Field(
        description="A brief summary of what the user is trying to accomplish"
    )
```

Entire Code Example

```python
from pydantic import BaseModel
from kura import Kura
from kura.summarisation import SummaryModel, ConversationSummary, GeneratedSummary
from kura.types import Conversation, Message
from typing import Literal
from pydantic import Field
import instructor

# Define our Response Model
class UsagePattern(BaseModel):
    category: Literal["Learning", "QuestionAnswering", "Brainstorming", "TaskCompletion"] = Field(
        description="The primary usage pattern of this conversation"
    )
    summary: str = Field(
        description="A brief summary of what the user is trying to accomplish"
    )

class UsagePatternModel(SummaryModel):
    def __init__(self):
```

Using our new model

```python
from kura import Kura

kura = Kura(
    summarisation_model=UsagePatternModel()
)
```

Understanding RAG Systems - Inventory vs Capability

Language Detection for Conversations

```python
import asyncio
import instructor
from pydantic import BaseModel, Field

class Language(BaseModel):
    language_code: str = Field(
        description="The language code of the conversation. (Eg. en, fr, es)",
        pattern=r"^[a-z]{2}$",
    )
```

Entire Code Example

```python
import asyncio
import instructor
from pydantic import BaseModel, Field
from kura import Kura
from kura.types import Conversation, ExtractedProperty
from kura.summarisation import SummaryModel


class Language(BaseModel):
    language_code: str = Field(
        description="The language code of the conversation. (Eg. en, fr, es)",
        pattern=r"^[a-z]{2}$",
    )


async def language_extractor(
    conversation: Conversation,
    sems: dict[str, asyncio.Semaphore],
    clients: dict[str, instructor.AsyncInstructor],
```

Make Your Customer Data Work For You

Start making data-driven decisions today

Convert unstructured conversations into actionable insights in minutes

usekura.xyz