Case Study: Data Engineering Methodologies in Telecom

Introduction

In this case study, we will explore the dynamic world of data engineering methodologies within the telecom industry. Our focus will be on understanding data sourcing patterns, mastering data modelling techniques, and embracing various consumption mechanisms to enhance decision-making processes. By delving into batch and stream ingestion patterns, curated data modelling, and diverse consumption avenues, we aim to equip graduates with practical insights into the realm of data engineering.

Dataset Selection

To facilitate our exploration, we will harness the potential of a diverse range of telecom-related datasets, each offering a unique perspective on the industry:

1. **Telecom Call Records:** A dataset capturing call logs, including caller and receiver details, call duration, and timestamps.
2. **Customer Usage Data:** This dataset sheds light on individual customer data usage, encompassing voice minutes, text messages, and data consumption.
3. **Billing and Payment History:** A dataset chronicling billing information, payment dates, and outstanding balances for each customer.
4. **Network Performance Metrics:** A collection of data spotlighting network performance aspects, encompassing signal strength, call drop rates, and data transfer speeds.
5. **Customer Complaints:** A dataset meticulously recording customer complaints, categorizing them, detailing resolution times, and assessing customer satisfaction.

Data Sourcing Patterns

Batch Ingestion Pattern

Scenario 1: Extracting Call Records

Objective: Source call records data using the batch ingestion pattern for further analysis and billing purposes.

Steps:

1. Obtain telecom call records data from different sources (e.g., network switches).
2. Apply batch ETL processes to transform and clean the data.
3. Store the transformed data in a centralized storage system (e.g., data warehouse, data lake).

Schema:

- **Call Records:** (Caller, Receiver, CallStart, CallDuration)

Stream Ingestion Pattern

Scenario 2: Real-time Network Monitoring

Objective: Collect real-time network performance metrics to monitor signal strength, call drop rates, and data transfer speeds.

Steps:

1. Stream network performance data from various devices (e.g., routers) using streaming frameworks (e.g., Apache Kafka).
2. Implement real-time processing to enrich and analyze the data.
3. Trigger alerts for network anomalies and performance degradation.

Schema:

- **Network Performance:** (DeviceID, Timestamp, SignalStrength, CallDropRate, DataTransferSpeed)

Data Modelling Techniques

Curated Data Model

Scenario 1: Unified Customer View

Objective: Create a unified view of customers by integrating data on usage, billing, network performance, and complaints for improved decision-making.

Steps:

1. Identify key entities such as customers, usage, billing, network performance, and complaints.
2. Design a star schema for the curated data model.
3. Build dimension tables for customers, time, location, etc.
4. Construct fact tables for usage, billing, network metrics, and complaints.
5. Populate the curated data model using ETL processes.

Schema:

- **Customers:** (CustomerID, Name, Address, ContactInfo, SubscriptionDate, ChurnStatus)

CustomerID	Name	Address	ContactInfo	SubscriptionDate	ChurnStatus
b019d218-4f18-4ea1-8c15-93f72b05df72	John Doe	123 Main St	555-123-4567	2023-01-15	Active
377ddade-8e97-47b6-8e58-547ef7a8c50a	Jane Smith	456 Elm Ave	555-987-6543	2023-02-10	Active

- **Time:** (Date, Month, Year)

Date	Month	Year
2023-08-01	August	2023
2023-07-20	July	2023

- **Usage:** (CustomerID, Date, CallMinutes, DataUsage, TextMessages)

CustomerID	Date	CallMinutes	DataUsage	TextMessages
b019d218-4f18-4ea1-8c15-93f72b05df72	2023-08-01	120	2500	80
377ddade-8e97-47b6-8e58-547ef7a8c50a	2023-07-20	80	1200	40

- **Billing:** (CustomerID, Month, BillingAmount, PaymentDate, OutstandingBalance)

CustomerID	Month	BillingAmount	PaymentDate	OutstandingBalance
b019d218-4f18-4ea1-8c15-93f72b05df72	August	125.45	2023-08-10	12.70

377ddade-8e97-47b6-8e58-547ef7a8c50a	July	78.20	2023-07-25	0.00

- **NetworkMetrics:** (CustomerID, Date, SignalStrength, CallDropRate, DataTransferSpeed)

CustomerID	Date	SignalStrength	CallDropRate	DataTransferSpeed
b019d218-4f18-4ea1-8c15-93f72b05df72	2023-08-01	85	2.3	56
377ddade-8e97-47b6-8e58-547ef7a8c50a	2023-07-20	70	1.5	42

- **Complaints:** (CustomerID, Date, ComplaintType, ResolutionTime, SatisfactionRating)

CustomerID	Date	ComplaintType	ResolutionTime	SatisfactionRating
b019d218-4f18-4ea1-8c15-93f72b05df72	2023-08-05	Network Issue	3	3
377ddade-8e97-47b6-8e58-547ef7a8c50a	2023-07-22	Billing Dispute	2	4

Consumption Mechanisms

Generating Downstream Feed File

Scenario 1: Monthly Billing Reports

Objective: Generate and distribute monthly billing reports to customers, facilitating transparent billing communication.

CustomerID	Month	BillingAmount	PaymentDate	OutstandingBalance
b019d218-4f18-4ea1-8c15-93f72b05df72	August	125.45	2023-08-10	12.70
377ddade-8e97-47b6-8e58-547ef7a8c50a	August	78.20	2023-08-15	0.00
a6c49b54-6b14-46cf-955b-42f4b8e6bf7d	August	98.75	2023-08-05	25.35
b019d218-4f18-4ea1-8c15-93f72b05df72	July	112.80	2023-07-15	5.90
377ddade-8e97-47b6-8e58-547ef7a8c50a	July	85.40	2023-07-20	0.00
a6c49b54-6b14-46cf-955b-42f4b8e6bf7d	July	103.25	2023-07-10	17.15

Steps:

1. Extract relevant data from the curated data model.
2. Transform the data into the required format (e.g., CSV, Excel).
3. Automate the process to generate reports monthly.
4. Distribute the reports to customers via email or file-sharing platforms.

Building Reports and Insights in Power BI

Scenario 2: Network Performance Dashboard

Objective: Develop a Power BI dashboard to visualize and analyse network performance metrics for optimization.

Steps:

1. Connect Power BI to the curated data model.
2. Design interactive visualizations for network metrics.
3. Incorporate filters and slicers for detailed analysis.
4. Schedule regular data refresh to maintain dashboard accuracy.

Sample Network Performance Dashboard Visuals

- **Signal Strength Trend:** A line chart showing the trend of signal strength over time. The X-axis represents time (e.g., days, weeks), and the Y-axis represents signal strength values. Different lines could represent signal strength for different customers.
- **Call Drop Rate Analysis:** A bar chart displaying the call drop rates for different customers. Each bar represents a customer, and the height of the bar represents the call drop rate percentage.
- **Data Transfer Speed Comparison:** A grouped bar chart comparing the average data transfer speeds for different months. Each group represents a month, and bars within the group represent the average data transfer speeds for different customers.
- **Geographical Heatmap:** A geographical heatmap displaying signal strength variations across different regions. Darker areas indicate stronger signal strength, while lighter areas indicate weaker signal strength.
- **Network Anomalies Alert:** A real-time alert widget showing a notification whenever the call drop rate exceeds a predefined threshold. This could be in the form of a blinking indicator or a pop-up alert.
- **Customer Complaints Analysis:** A donut chart displaying the distribution of different complaint types reported by customers. Each slice represents a complaint type, and the size of the slice corresponds to the percentage of complaints.
- **Customer Satisfaction Ratings:** A scatter plot showing the relationship between customer satisfaction ratings and call drop rates. Each point represents a customer, with the X-axis representing call drop rate and the Y-axis representing satisfaction rating.
- **Network Performance Overview:** A summary card displaying key metrics such as average signal strength, overall call drop rate, and average data transfer speed for the selected time period.
- **Time-based Filter:** A slider or calendar picker allowing users to dynamically adjust the time range of data displayed on the dashboard.

Building APIs for Digital Channel Consumption

Scenario 1: Customer Usage API

Objective: Create an API to empower customers to access their usage data for personalized insights.

Steps:

1. Design API endpoints to retrieve usage data for specific customers.
2. Implement secure authentication and authorization mechanisms.
3. Fetch data from the curated data model based on API requests.
4. Transform and format the data into JSON for API responses.
5. Apply API rate limiting and security measures.

Sample Request Payload:

```
GET /api/customer-usage?customerID=b019d218-4f18-4ea1-8c15-93f72b05df72&startDate=2023-07-01&endDate=2023-07-31
```

In this example, the client is requesting customer usage data for a specific customer (customerID), within a specified date range (startDate to endDate).

Sample Response Payload:

```
{
  "customerID": "b019d218-4f18-4ea1-8c15-93f72b05df72",
  "usageData": [
    {
      "date": "2023-07-01",
      "callMinutes": 120,
      "dataUsage": 2500,
      "textMessages": 80
    },
    {
      "date": "2023-07-15",
      "callMinutes": 90,
      "dataUsage": 1800,
      "textMessages": 60
    },
    {
      "date": "2023-07-25",
      "callMinutes": 150,
      "dataUsage": 3200,
      "textMessages": 100
    }
  ]
}
```

The API response includes the requested customer's customerID and an array of usage data records for the specified date range. Each usage data record contains the date, callMinutes, dataUsage, and textMessages fields.

Python Script for Generating Connected Data

Here's an example Python script to generate connected data for the telecom datasets:

synthetic_data_generator.py   synthetic_data_generator_v2.0.py

Please note that this script is a simplified example and may require further customization and integration into a larger data engineering pipeline. You can adapt and expand upon this script to generate more complex and realistic connected data for your case study.

This structured case study is designed to provide graduates with an in-depth understanding of data engineering methodologies, enabling them to tackle real-world challenges in the telecom domain. By delving into various patterns, modelling techniques, and consumption mechanisms, graduates will be well-prepared to excel in the dynamic field of data engineering.