**Data Attribution System: Approach, Assumptions, Observations, and Future Scope**

**Introduction**

This document outlines the approach undertaken to build a simple attribution system for understanding which touchpoints and channels are sourcing the most pipeline. The system adheres to the specified properties and assumptions, including a 90-day attribution window and a first-touch attribution model.

**Approach**

1. **Import Data:**
   * Load data from the provided CSV and Excel files for contacts, marketing touchpoints, sales touchpoints, and opportunities.
2. **Merge Marketing and Sales Data:**
   * Combine marketing and sales touchpoints into a single DataFrame, differentiating them by a touchpoint\_type column.
3. **Join Touchpoint Data with Contact and Opportunity Data:**
   * Merge the combined touchpoint data with the contact data on contact\_id.
   * Merge the result with the opportunity data on account\_id.
4. **Filter Touchpoints within 90 Days Before Each Opportunity Creation:**
   * Calculate the days between each touchpoint and the corresponding opportunity creation date.
   * Filter out touchpoints that fall outside the 90-day window.
5. **Identify the First Touchpoint for Each Opportunity:**
   * Sort the filtered touchpoints by opportunity ID and touchpoint date.
   * Group by opportunity ID to select the first touchpoint within the 90-day window for each opportunity.
6. **Calculate Sourced Pipeline:**
   * Attribute the pipeline amount of each opportunity to the corresponding first touchpoint.
7. **Validate the populated data**
   * Validate the aggregated data to ensure there are no data discrepancies in the output.
8. **Create the Final Output Table:**
   * Select relevant columns and save the result to a CSV file for further analysis.
   * Here I am outputting a csv, but it can be loaded into a table.

**Assumptions**

1. **Account-Level Attribution:**
   * Interactions on an account are related to opportunities on that account.
2. **First Touch Attribution Model:**
   * The first touchpoint within the 90-day period before the opportunity creation date gets full credit for sourcing the opportunity.
3. **Multiple Opportunities per Account:**
   * An account can have multiple opportunities, and each opportunity should be considered independently for touchpoint attribution.

**Observations**

1. **One Account ID through a Contact Can Have Multiple Opportunity IDs:**
   * This scenario was observed and handled by ensuring that touchpoints are filtered and attributed independently for each opportunity.

**Future Scope**

1. **Enhanced Attribution Models:**
   * Implement additional attribution models such as last-touch, linear, or multi-touch to provide a comprehensive analysis.
2. **Cost Data Integration:**
   * Incorporate cost data for each touchpoint or channel to enable ROI analysis.
3. **Real-Time Data Processing:**
   * Develop a real-time data processing pipeline to handle live data streams and provide up-to-date attribution analysis.
4. **Automated Data Validation:**
   * Implement automated data validation and quality checks to ensure the accuracy and reliability of the insights.

**Answers to Questions**

1. **Which channel sourced the most pipeline? How does this look by sales segment?**

**Total Pipeline Sourced by Each Channel:**

Adwords: 4,374,556

Event: 2,274,761

Outbound: 33,978,975

Webinar: 3,228,655

Website: 4,698,794

**Total Pipeline by Sales Segment:**

Commercial: 12,933,116

Enterprise: 29,598,097

Mid Market: 6,024,528

**Outbound Pipeline Breakdown by Sales Segment:**

Commercial: 8,738,545

Enterprise: 21,098,564

Mid-Market: 4,141,866

* + **Channel sourcing the most pipeline:** **Outbound** channel had sourced the maximum opportunities with the total pipeline amount of 33, 978,975
  + The **Outbound** channel sourced the most pipeline overall. The breakdown by sales segment shows that the Enterprise segment contributed the most to the Outbound channel's pipeline, followed by Commercial and Mid-Market segments. ​​

1. **What information do you need to know to understand the ROI (return on investment) of each channel?**
   * **Cost data:** The cost associated with each marketing or sales touchpoint/channel.
   * **Revenue data:** The actual revenue generated from opportunities sourced by each channel.
   * **Conversion rates:** The rate at which leads from each channel convert into opportunities and then into closed deals.
2. **How did you structure your data table and why? What do you think are the important output dimensions?**

The data table is structured with the following columns:

touchpoint\_id, channel\_name, contact\_id, account\_id, opportunity\_id, touchpoint\_date, Opportunity\_Created\_Date, pipeline\_amount, Sourced\_Pipeline, sales\_segment

Reasoning**:** This structure was chosen for several key reasons:

**Granularity**: Each row represents a single touchpoint that led to an opportunity, allowing for detailed analysis at the most granular level.

**Traceability**: By including both touchpoint and opportunity information, we can trace the entire journey from initial interaction to opportunity creation.

**Hierarchical representation**: The inclusion of contact\_id, account\_id, and opportunity\_id allows for analysis at different levels of the sales hierarchy (individual, company, and deal levels).

**Time-based analysis**: Including both touchpoint\_date and Opportunity\_Created\_Date enables time-based analyses, such as measuring the time from first touch to opportunity creation and understanding seasonality.

**Attribution clarity**: The Sourced\_Pipeline column clearly shows the attributed amount for each touchpoint, making it easy to aggregate and analyze attribution data.

**Important output dimensions:**

**Channel name:**

Importance: Allows for analysis of which channels are most effective in sourcing pipeline.

Use cases: This field helps in assessing channel performance comparison, budget allocation decisions, understanding the most effective touchpoints in the customer journey.

**Sales segment:**

Importance: Enables segmentation of data by different market segments or customer types.

Use cases: It helps in tailoring strategies for different segments, identifying which channels work best for each segment, allocating resources based on segment performance.

**Touchpoint date and Opportunity Created Date:**

Importance: Enable time-based analysis and trending.

Use cases: Using this field we can do seasonal trend analysis, measuring time-to-opportunity for different channels, understanding the sales cycle length across segments and channels.

**Account ID and Opportunity ID:**

Importance: Allow for account-based and opportunity-level analysis.

Use cases: We can use it for account-based marketing strategies, understanding complex B2B sales cycles, analyzing multi-touch attribution within accounts.

**Contact ID:**

Importance: Enables individual-level analysis.

Use cases: It helps in understanding individual customer journeys, personalization strategies, analyzing the impact of multiple contacts within an account.

**Important output measures:**

**Sourced Pipeline:**

Importance: This is the key metric for measuring the impact of each touchpoint.

Use cases: ROI calculations, performance evaluations of different channels and campaigns, determining the most influential touchpoints.

**Pipeline Amount:**

Importance: Represents the total opportunity value.

Use cases: Comparing attributed amount to total opportunity value, understanding deal sizes across segments and channels.

This structure and these dimensions/measures allow for a wide range of analyses, including:

1. Channel effectiveness across different segments and time periods
2. Account-based insights and multi-touch attribution within accounts
3. Time-based trends and seasonality in pipeline generation

Funnel analysis from initial touchpoint to opportunity creation

By including these dimensions and measures, we create a flexible dataset that can answer a variety of business questions and support different analytical approaches. It balances the need for detailed, granular data with the ability to easily aggregate and summarize for high-level insights, making it valuable for both operational teams and executive decision-making.

1. **This table is an important input into other data and business systems. What kind of data validations and checks would you implement to make sure that downstream stakeholders have confidence in the insights they are generating from this?**
   * **Date range checks:** Ensure touchpoints fall within the valid 90-day window before the opportunity creation date.
   * **Duplicate checks:** Ensure no duplicate touchpoints or opportunities.
   * **Null checks:** Ensure critical columns (e.g., channel\_name, opportunity\_id, pipeline\_amount) do not contain null values.
   * **Consistency checks:** Verify touchpoint dates are consistent with opportunity creation dates and that the touchpoints belong to the correct accounts.
   * **Aggregated data validation:** Cross-verify the aggregated pipeline amounts with the source data to ensure accuracy.
   * **Unit tests:** Implement unit tests for each function to validate expected transformations and calculations.

How would you think about architecting a robust attribution system? A ‘robust attribution

system’ should, at minimum, be able to do the following:

➔ Ingest, transform and map interaction and opportunity data from several sources

➔ Apply several different attribution models with unique logic

➔ Pipe outputs into an easily accessible and usable system for end stakeholder

consumption. End stakeholders include executive leadership, sales, marketing

and finance.

➔ Repeat the above cycle on a daily or more frequent basis

Please answer the following questions:

➔ What data platforms will you use and what will your data stack look like? Why?

➔ Please identify risks and vulnerabilities in your system

➔ How will you maintain and scale this system?

1. Data Ingestion and Storage:

**Streaming data**

I would use Apache Kafka for real-time data streaming, enabling high-throughput, fault-tolerant, and horizontally scalable event processing. Kafka's distributed nature allows it to handle millions of events per second, making it ideal for capturing user interactions and touchpoints in real-time.

**Batch ingestion**

Airbyte/Fivetran for batch data ingestion from various sources, providing a flexible and extensible open-source platform that can connect to hundreds of data sources and destinations. Airbyte's ability to handle both structured and unstructured data makes it valuable for integrating diverse data types into the attribution system.

**Storage**

Amazon S3 or Google Cloud Storage as a data lake for raw data storage, offering virtually unlimited scalability, high durability, and cost-effective storage options. These cloud-based object storage services provide a centralized repository for all raw data, enabling easy access and processing by downstream systems.

1. **Data Processing, Transformation, and Storage:**
2. I would use Databricks for large-scale data processing, analytics, and machine learning. As databricks provides a unified analytics platform that combines the best of data warehouses and data lakes into a lakehouse architecture. It offers:
   * Collaborative notebooks for data exploration and analysis
   * Delta Lake for ACID transactions on data lakes
   * MLflow for end-to-end machine learning lifecycle management
   * SQL Analytics for running SQL queries directly on the data lake
   * Optimized Apache Spark runtime for improved performance
3. dbt (data build tool) for data transformation and modeling, enabling analytics engineers to write, document, and execute data transformations using SQL. When used with Databricks:
   * dbt can run transformations directly on Databricks clusters
   * It leverages Databricks' optimized Spark engine for efficient processing
   * Enables version-controlled, modular data transformations
   * Provides data lineage and documentation features
   * Allows for testing and validation of data models

**Workflow Orchestration**:

For orchestration, I would use Apache Airflow for orchestrating and scheduling data pipelines, providing a programmatic approach to authoring, scheduling, and monitoring workflows. Airflow's rich ecosystem of operators and hooks allows seamless integration with Databricks and other components of the attribution system, enabling complex multi-step data pipelines.

Attribution Modeling:

As attribution modeling using a customized logic, I would use custom Python/Scala modules for implementing various attribution models, allowing for flexibility in designing and refining attribution logic. These custom modules can be developed and run within Databricks notebooks, taking advantage of:

* Distributed computing capabilities for processing large datasets
* Integration with MLflow for tracking experiments and model versions
* Easy access to historical data stored in Delta Lake format
* Collaboration features for data scientists and analysts

Serving Layer:

1. GraphQL API using Apollo Server for flexible data querying, providing a single endpoint for clients to request exactly the data they need. GraphQL's strong typing and self-documenting nature make it easier for front-end developers to consume attribution data efficiently.
2. Redis for caching frequently accessed data, reducing latency and database load for commonly requested information. Redis's in-memory data structure store can significantly improve query performance for real-time dashboards and reports.

Visualization and Reporting:

Tableau or Looker for creating interactive dashboards and reports, enabling stakeholders to explore attribution data visually and derive actionable insights. These BI tools can connect directly to Databricks SQL endpoints, allowing for:

* Real-time querying of the latest attribution data
* Creation of interactive visualizations and dashboards
* Self-service analytics for business users
* Scheduled report generation and distribution

Monitoring and Alerting:

1. Prometheus and Grafana for system monitoring, providing real-time visibility into system performance metrics and enabling proactive issue detection. These tools can be used to monitor Databricks clusters, Kafka streams, and other components of the attribution system.
2. Datadog for application performance monitoring, offering end-to-end tracing, log management, and infrastructure monitoring. Datadog's ability to correlate metrics, traces, and logs helps in quickly identifying and resolving performance bottlenecks across the entire attribution system stack, including Databricks jobs and clusters.

This revised architecture leverages Databricks as a central platform for data processing, storage, and analytics, while still incorporating other best-in-class tools for specific functions. The combination of Databricks and dbt provides a powerful, flexible, and scalable solution for building and maintaining complex attribution models and data pipelines.