Data Science Exam Report

Part 1: Database and Data Understanding

I.e: Found in data_exploration.ipynb

1. Complex Querying

Query 1: Fetch the top 10 users who contributed the highest revenue within the last 30 days, along with the number of groups they participated in and the categories of products they purchased.

- This query will join the `users`, `orders`, `groups_carts`, `groups`, and `product_names` tables to derive the desired information.
- **Purpose**: Identify high-value users to inform targeted marketing and reward programs.

Query 2: Calculate the conversion rate for group deals.

- This query uses the `group_deals` and `orders` tables, filtering orders with a status of 'completed' and dividing them by all group deals created.
- **Purpose**: Measure the effectiveness of group deals in driving completed transactions.

Part 2: Data Processing

2. Advanced Data Aggregation

Monthly Cohort Analysis:

- A Python script will fetch data from the `users` and `groups` tables, grouping users by their signup month (`created_at`) and calculating retention percentages based on group deal participation over the next three months.
- **Purpose**: Track user retention trends and evaluate engagement strategies.

3. Dynamic Data Preparation

Popular Product Categories and Sales Growth:

- A Python function will dynamically fetch data from `products` and `orders`, calculating the growth percentage of product categories over defined periods.

Reusable Data Preprocessing Class:

- A Python class will preprocess all tables, handling NULL values, encoding categorical variables, and aggregating timestamps.
- **Purpose**: Standardize and streamline data preparation for various analyses.

Part 3: Dashboard Development

For the Streamlit project, navigate to: `ChipChip\streamlit_app>tree /f`

This directory contains:

1. **app.py**: The main Streamlit application file, where the app logic is defined.
2. **preprocessing**: A folder that includes:
    - **data_preprocessor.py**: A script for data cleaning, transformation, and preparation.
    - **pycache**: A directory storing Python bytecode for faster execution.

In the Streamlit dashboard, you'll find:

- **Heatmap** showing correlations between product categories and vendor order contributions.
- **Time-series visualizations** using ARIMA or Prophet for order trend forecasting.
- **Grouped bar charts** comparing order quantities for group vs. individual deals.

Interactive features include:

- **Dynamic Filtering System** for product list updates based on vendor selection.
- **Multi-Select Widget** for comparing performance metrics like revenue and conversion rates.

Real-time data updates fetch new orders every 5 minutes, ensuring metrics stay current and anomalies are quickly identified.

Part 4: KPIs

KPI 1: Group Deal Conversion Rate

- **Definition**: Percentage of group deals that result in completed orders.
- **Importance**: Measures the success of group deals in driving actual sales.
- **SQL Query**:

```
SELECT
  (COUNT(CASE WHEN o.status = 'completed' THEN 1 END)::DECIMA
L / COUNT(*)) * 100 AS conversion_rate
FROM group_deals gd
JOIN orders o ON gd.id = o.groups_carts_id;
```

KPI 2: Average Revenue Per Group Deal

- **Definition**: Average revenue generated per group deal.
- **Importance**: Assesses the financial impact of group deals.
- **SQL Query**:

```
SELECT
  AVG(o.total_amount) AS avg_revenue
FROM group_deals gd
JOIN orders o ON gd.id = o.groups_carts_id;
```

KPI 3: User Participation Rate in Group Deals

- **Definition**: Proportion of users participating in group deals.
- **Importance**: Evaluates user engagement with group deals.
- **SQL Query**:

```
SELECT
  (COUNT(DISTINCT gc.user_id)::DECIMAL / (SELECT COUNT(*) FRO
```

```
M users)) * 100 AS participation_rate
FROM groups_carts gc;
```

Part 5: Data Engineering Challenge

Real-Time Reporting for ChipChip

For ChipChip, The following business priorities and strategies are recommended to design and maintain a real-time reporting system tailored to ChipChip’s unique business model:

1. Enhance Group Deal Tracking

- **Why Prioritize**: The group deal mechanism is the platform's core. Real-time tracking of group statuses (active, completed, or expired) directly impacts customer experience, refund processing, and marketing decisions.
- **Action Steps**:
  - Build ETL pipelines to extract, transform, and load group deal data in real-time.
  - Use triggers in the database to flag deals nearing expiration or those achieving their member goals for instant notification.
  - Integrate visual dashboards showing group statuses across timeframes to guide marketing interventions, like offering reminders or incentives.

2. Optimize Refund Processing

- **Why Prioritize**: Refunds for expired groups must be processed seamlessly to maintain trust and customer satisfaction.
- **Action Steps**:
  - Automate refund workflows with ETL processes to handle expired groups as soon as they are flagged.
  - Regularly audit refund data pipelines to ensure accuracy.

- Develop anomaly detection models to identify potential refund delays or discrepancies.

---

3. Drive Customer Engagement Through Analytics

- **Why Prioritize**: Understanding customer behavior and engagement patterns is vital for personalizing deals and maximizing participation.
- **Action Steps**:
  - Design ETL pipelines to aggregate customer activity, including participation frequency, group success rates, and average spending.
  - Use these insights to tailor targeted marketing campaigns, like recommending group deals to high-value customers or engaging occasional buyers.

---

4. Enable Product and Vendor Performance Insights

- **Why Prioritize**: Identifying top-performing product categories and vendors can help optimize inventory and vendor relationships.
- **Action Steps**:
  - Build ETL pipelines to calculate sales contributions by product and vendor in near real-time.
  - Use correlation analysis to uncover trends, such as how specific product categories perform under varying group conditions.
  - Present these insights in dashboards, enabling data-driven decisions for stock replenishment and vendor incentives.

---

5. Support Demand Forecasting and Capacity Planning

- **Why Prioritize**: The group-buying model's time-sensitive nature necessitates accurate demand forecasting to avoid missed opportunities or wasted resources.
- **Action Steps**:

- Implement ETL processes that incorporate historical sales and user activity to train predictive models.
- Deploy dashboards to visualize demand forecasts, helping to adjust marketing spend or inventory availability dynamically.

---

6. Real-Time Monitoring for Operational Efficiency

- **Why Prioritize**: Operational challenges, such as slow loading times or delayed notifications, can negatively impact user trust and platform adoption.
- **Action Steps**:
  - Use ETL pipelines to generate operational health metrics, such as API response times and order processing delays.
  - Implement alerting systems that notify administrators when operational thresholds are breached.

Key Technical Considerations for Real-Time Reporting

1. **ETL Pipeline Design**
   - Use a hybrid batch-streaming model: Batch ETL for less time-sensitive data like historical sales trends and streaming ETL for real-time metrics like active group statuses.
   - Use tools like Apache Kafka or AWS Kinesis for streaming data ingestion.
2. **Query Optimization**
   - Partition group deal and order tables by time (e.g., daily partitions) for faster query execution.
   - Leverage indexes on key fields such as `id`, `status` to improve query performance.
3. **Data Consistency**
   - Use database transactions to ensure consistency when processing refunds or updating group statuses.

- Implement data validation at each stage of the ETL pipeline to catch and resolve anomalies early.

4. **Scalability**
    - Use cloud-based data warehouses like Amazon Redshift or Snowflake to handle growing data volumes.
    - Build a modular ETL architecture that can scale with additional data sources, such as new product categories or vendor types.
