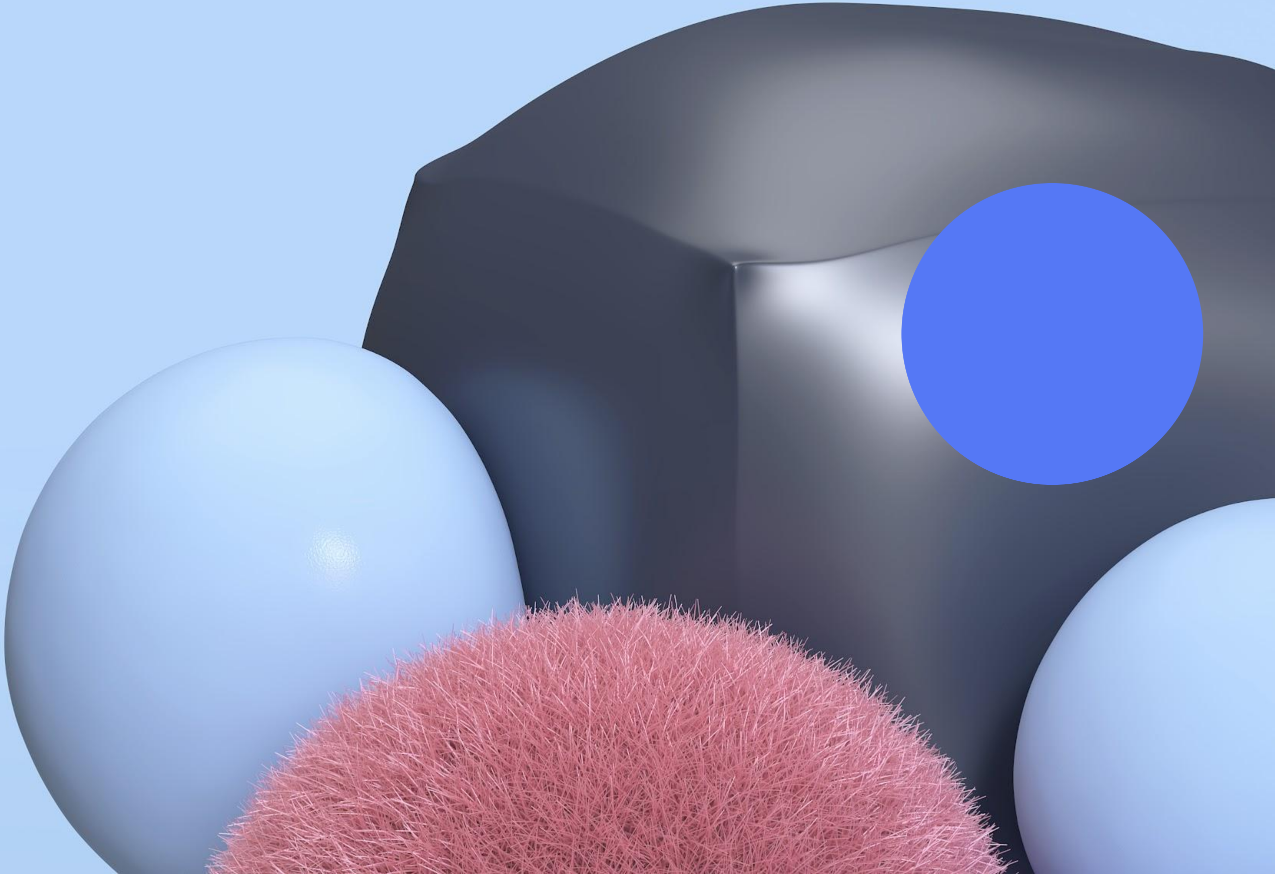
precis.

ML Engineer

CASE
MATERIAL



Guidelines

In this document, you'll be presented with some challenges to solve and subsequently present at your next interview with Precis.

Some general guidelines:

- You may choose to demo and/or present part 1 and 2. Presentation and code should be submitted by 11AM on the same day as your interview, to the people at Precis you will be meeting with.
- Your presentation should be in english.
- This case covers common use case for cloud-based applications at Precis. Prepare to apply knowledge of cloud services to design an infrastructure fully executable using the Google Cloud Platform.

Note
You can find contact details to the Precis people that you will meet for your interview in the meeting invite that has been sent out.

Case

The case consist of two parts which should be solved independent of each other.

Part 1

~20 min

Task

- Access the data using this Drive folder.

- Read in the datasets. Use the data dictionary on the next page to understand what each field represents.
  - Build a pipeline to ingest data as it is updated in the Drive folder.
  - Perform data preprocessing/cleaning up of the data.
  - Join the appropriate tables (wherever required) and process the following metrics:
    - What are the top product categories most commonly purchased by first-time customers?
    - How large is the segment of customers with an average payment value over 200 SEK?
    - Does the order value increase as the customers place more orders?

- Share the code and documentation as a private gitlab or github repo. Setup BigQuery for free here or use a notebook.

Data Dictionary

Customer

Column	Type	Description
customer_id	STRING	Unique identifier for a customer.
customer_city	STRING	City of the customer.
channel	STRING	The channel that acquired the customer.

Payment

Column	Type	Description
orderr_id	STRING	Unique identifier for an order.
payment_types	INTEGER	More than 1 if customer paid with different methods.
payment_method	STRING	Payment method.
payment_value	FLOAT	Order value.
currency	STRING	Currency of the payment.

Order

Column	Type	Description
order_id	STRING	Unique identifier for an order.
customer_id	INTEGER	Unique identifier for a customer.
order_status	STRING	Status of the order.
order_purchase_timestamp	TIMESTAMP	When customer made the order.
order_approved_at	TIMESTAMP	When order was approved.
order_delivered_carrier_date	TIMESTAMP	When order was delivered.
order_item_id	INTEGER	ID of the order item (1-99).
product_id	STRING	Unique identifier for a product.
seller_id	STRING	Unique identifier for a seller.
price	FLOAT	Product price.
freight_value	FLOAT	Freight value.
Product_*	STRING/ FLOAT	Different product meta data.

Part 2

Background

A large e-commerce client is calculating Customer Lifetime Value using aggregated revenue statistics from all customers. This year, they want to challenge their calculation with a more sophisticated model to help them better understand their customers' long-term value for their business.

Their client doesn't have all their data stored in one place and are missing the right skills in order to enable this dataflow internally.

The team has asked for your help to present a solution that would enable the following deliverables:

- Consolidation of their web behavior data (source: Google Analytics 360), customer data (source: Emarsys), and financial data (Source: Amazon S3) in Google Cloud Platform
- Present a suggestion on how to calculate Customer Lifetime Value in a better way.
- Export the data to Google Analytics and Google Ads daily.

Please find outlined tasks on the next page.

~30 min

Task

Your task is to define:

- A solution on the Google Cloud Platform that helps the client process their business data and activate the results in Google Analytics.

- Explain how you will manage the following:
    - What features would be relevant for developing such a model?
    - How would you handle feature engineering / data quality issues?
    - How would you evaluate the performance of the model and if it's suitable for production?
    - Do you see any risks or issues with the data?
    - What if we have only 1 or 2 features? What sort of model is suitable?
    - How do you handle data skew (e.g. large outliers, etc)?
    - How do you handle class imbalance?
    - What techniques would you use to communicate the effectiveness of the model to outside stakeholders?
    - What's your approach to regularization (overfit/underfit)?

- A rough timeline from build to deploy, including the key steps.

Good luck!