Stakeholder Questions to Consider

Demand Prediction - Machine Learning Team

Chris Chen
Tony Reina
Kyle Shannon
Suman Gunnala
Anil Luthra

Q1 What are the top 3 categories of books that are most read around Christmas?

Predicted sales could be considered a proxy for ‘read’ if we assume number of books read to be the volume of books moved subtracted from inventory.

We assume 'around Christmas' to mean one of the following:

- {Dec}
- {Nov, Dec}
- {Dec, Jan}
- {Nov, Dec, Jan}

In either of these cases, our algorithm already provides predictions for categories by month. Our prediction will be able to tell our customer what the top predicted categories are for each of the above time segments.

To determine what the top categories are for each of the above time periods from a historical purpose, then this is a simple SQL query on the Postgres data source, or a query along the mediated schema and aggregated by month.

As a Database Lookup

Determining the top categories for holiday seasons is a relatively simple database look up. One such query is displayed below. What the business is really asking for is what are the categories that seem to do really well around the holidays. And if those can be determined, then the machine learning question would naturally follow as what is the predicted volume we should stock to meet the demand, but not to have so much surplus as to waste money on keeping stock.

```
-- Q1
SELECT category, sum(books_sold) AS num_sold FROM monthly_sales
WHERE mon = 11 or mon = 12
GROUP BY category
ORDER BY num_sold DESC
LIMIT 3
```

As a Predictive Task

Certainly our customer could reasonably ask for a specific model that would focus on *Christmas* time. By querying for data that follows in that specific time frame we could more finely tune a model to learn about the variance present in that specific data. by doing this we would have a separately trained model that we would use for the Christmas holiday season.

We would supply a dataLog query and be retuned back training data for the months we care about. We would train and validate our model, persist it, save the training data used to a separate table, and then use the same query to inject new test data. Predictions will be written to a predictions table for the customer to use. Similarly we can build a dashboard using the predictions table. Providing a usable interface for the customer.

Here is what a dataLog query would look like, we omitted the body because at this time there is no mediated schema to query upon. This will be the case for all following dataLog queries in each successive question.

```
Q1(nodeID, count_orderID) :-
```

Q2 What time of the year are the sales of “Education” books the highest?

We can interpret this question in two ways.

1. A Lookup via the mediated schema for historical trends. Thereby determining via a plot/statistics if there is a ‘hot’ demand time of the year or if it is more flat.

2. Predictive task whereby for category of ‘education’ we predict the demand for each month and determine which month we assume will have the most volume in sales.

As a Database Lookup

The customer can look at historical data to determine if e.g. September is a good month to stock up on more educational books. There are many nodeIDs that include the word "Education".

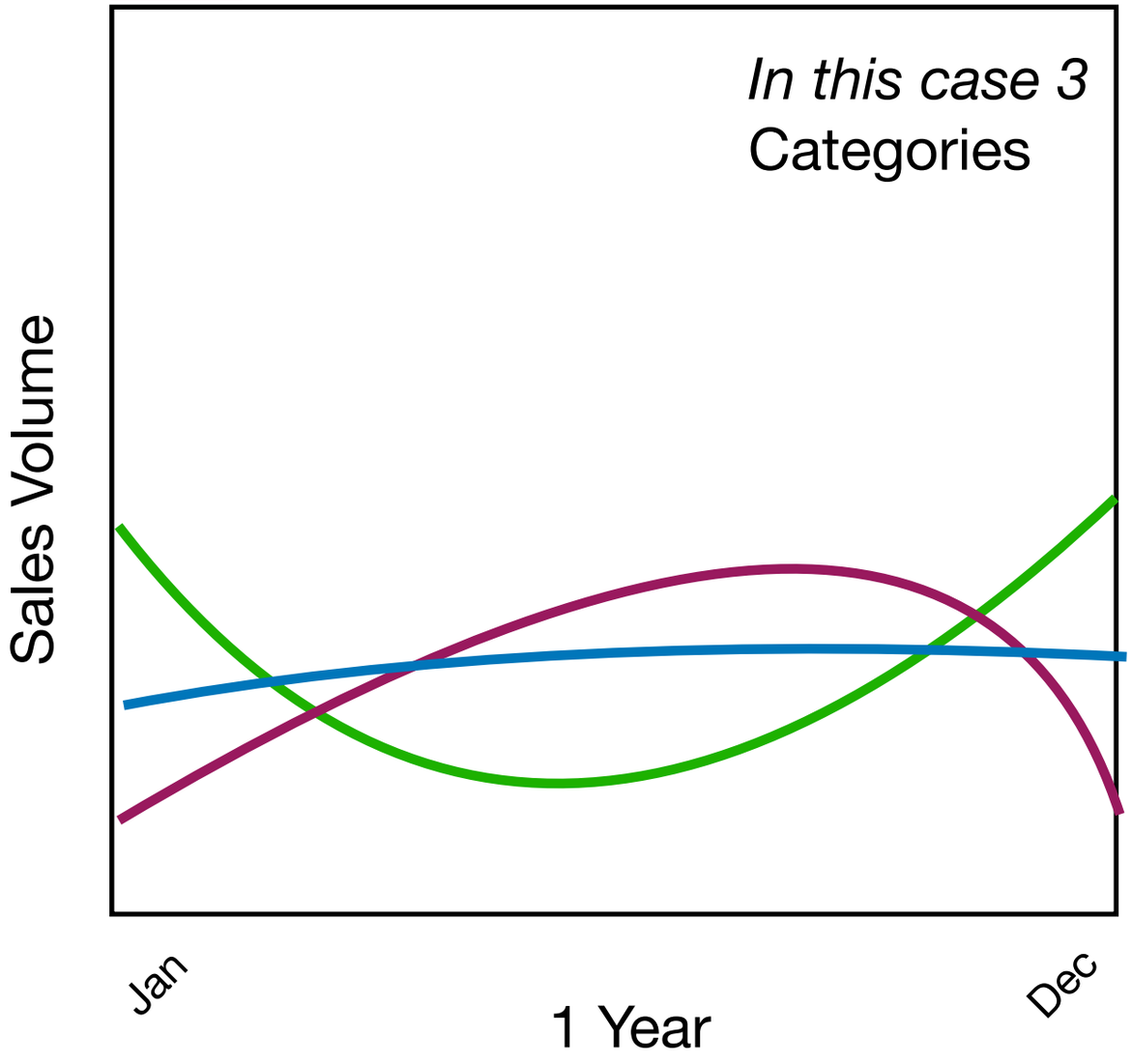
The query team knows that they need to create a way to look up nodeIDs based on a word search. For example, if we say "Give us all of the nodeIDs where the word "Education" shows up on any level of classification". So it's a JSON query to get these nodeIDs. Then from there we have a way to connect to the Asterix db for sales.

A database lookup would be a query that would closely match the previous SQL query from Q1. We want the sales data aggregated for a category across months. From there it is a matter of EDA/plotting/statistics to determine if there is periodicity to the signal for educational sales throughout the year. In the time domain is there a seasonal variance. If so can we capture it, which should be easy to find in a plot. Two example plots are drawn on the next slide. One for *n* categories over a year, and the other for all years for a given category.

This sample plot shows what could be the performance of 3 categories over a given year. Here we see that Purple does well in fall, Green has a holiday variance we might be able capture and Blue does fairly well over the whole year.

P1 *n* categories over 1 year

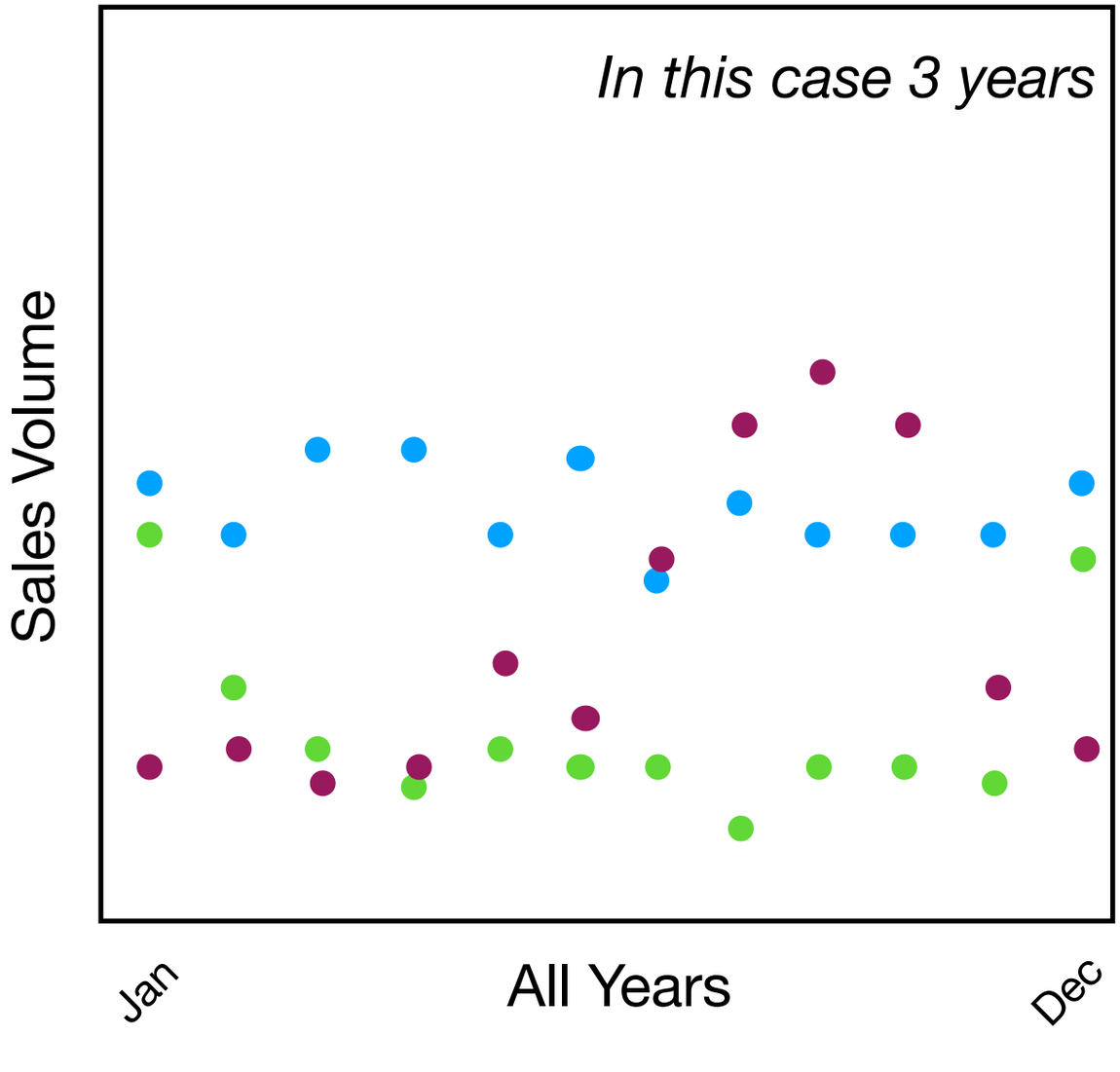
In this case 3
Categories



In this case Purple seems to have a *educational* component maybe ales are good in September area. Blue seems to always seem good, and Green has a winter holiday effect

P2 *n* years over 1 category

In this case 3 years



As a Predictive Task

Our current model already predicts for categories. The questions we need to ask is what does the customer mean by ‘time of year’ and what set of books constitute a category for ‘educational’? Our model is training and making predictions on a monthly scale.

If the client wanted to get less granular, say for the quarter this would be easy, we simply aggregate our predictions. If they wanted to get more granular then our model’s inputs would need to be adjusted slightly and retrained.

Below is the associated dataLog query.

```
4
5   Q2(month) :-
```

Q3 Given month m and category c, predict the amount of sales for the category.

This is the precise question our machine learning algorithm answers. The caveat being that our model is only currently being trained on the top 75 categories. In order to predict on any category, we must retrain on our model on every category. However we made this call due to the volume of book being moved. Below is the dataLog query.

```
6
7  Q3(date_agg_month, inventory_sold_ratio, dollar_sold_ratio,
8     volume_moved, product_rating_avg, product_rating_delta,
9     total_sales, contains_sold_out_product, large_inventory_drop,
10    is_pos_sentiment,is_neg_sentiment, is_neutral_sentiment,
11    count_of_nodeID, is_in_campaign) :-
```

Q4 Which book categories show a downward trend in demand in Winter and Spring?

Assuming Spring is defined as the period from February to May. We can determine the average change in the number of books sold from month-to-month during a three month windows.

If the result of the bottom query for a given category results in a negative number, then this shows a downward trend in volume moved for the category.

```sql
-- Spring
SELECT s.category, round(avg(s.change_in_sales_from_last_month)) AS sale_trend
FROM
(
SELECT category, mon, count(mon) OVER (PARTITION BY category) as num_months,
books_sold - lag(books_sold,1) over (PARTITION BY category ORDER BY mon) as change_in_sales_from_last_month
FROM monthly_sales
WHERE mon >= 2 AND mon <= 5
GROUP BY category, mon, books_sold
) AS s
WHERE s.num_months = (5-2) AND s.mon > 2 AND s.mon <= 5
GROUP BY s.category
ORDER BY sale_trend ASC

-- Winter

SELECT s.category, round(avg(s.change_in_sales_from_last_month)) AS sale_trend
FROM
(
SELECT category, mon, count(mon) OVER (PARTITION BY category) as num_months,
books_sold - lag(books_sold,1) over (PARTITION BY category ORDER BY mon) as change_in_sales_from_last_month
FROM monthly_sales
WHERE mon >= 8 AND mon <= 12
GROUP BY category, mon, books_sold
) AS s
WHERE s.num_months = (12-8) AND s.mon > 8 AND s.mon <= 12
GROUP BY s.category
ORDER BY sale_trend ASC
```

As a Predictive Task

Looking for a downward trend will not be an issue, so long as the time frame is agreed upon and set for a series of n months which constitute *spring* and *winter*. Once these guidelines are determined, then we simply query for the associated range train a model to predict volume over the given period and look for predictions that show a decrease in the volume for that category.

Here EDA can really help to answer the why question of such a prediction. Because there are going to be latent temporal factors that are present in the prediction space which a regression model has lashed onto. If you can identify those factors then you can begin to understand why a machine learning algorithm has picked up on them.

Luckily regression type models pick up on interactions that are more understandable and attainable through EDA then say a kernel trick where a decision boundary was determined in extremely high dimensional space.

Below is the associated dataLog query.

```
Q4_spring(nodeID, spring_sale_tend) :-
Q4_winter(nodeID, winter_sale_tend) :-
```

Q5 Is there a category that we should discontinue stocking?

To determine which category we should discontinue stocking, we must first establish a heuristic or threshold to follow. And from that decide which category falls below the threshold. We present one such example on the following slide.

One issue with a question such as this are the assumptions that we must make without more input from the stakeholders. Do they have a business rule for when a book should be discontinued? Are there performance KPIs that must be considered for the sales performance of each book. Furthermore do we need to look at the raw data to help determine if we can assist in development of such KPIs. Below is a SQL query to look up information that we could apply to a heuristic when deciding to discontinue a category.

```
1  CREATE VIEW yearly_sales AS
2  SELECT EXTRACT(YEAR from o.billdate) as yr,
3  p.nodeid as category, sum(o.numunits) as books_sold
4  FROM orderlines as o, products as p
5  WHERE o.productid = p.productid
6        AND
7        o.totalprice > 0::money
8  GROUP BY p.nodeid, EXTRACT(YEAR from o.billdate)
9  ORDER BY p.nodeid
```

One Method to Determine Discontinuing a Category

1. Create year sales view - using Orderlines.numunits for each nodeId/category of the book (Postgres DB sql based)
2. From the “yearly sales view“ Find categories whose total YEARLY sale is less than a threshold number (e.g.- for the last 5 year. Note here we are finding the categories which consistently were sold less in the last 5 years
3. (ML model usage) For each category (which is result of point 2) :
    1. Predict the demand for next year (or next quarter)
    2. If Demand continues to be less then we can discontinue the category
4. By the above two methods we use the historical sales data and prediction model (ML) to identify categories which can be discontinued.

Using a simple dataLog query, such as the one below, we could gets information needed about any given nodeID, or category, and by looking at the data we could build a separate model (if business needs required it) and try to predict when a category will begin to founder. If such a model performed very well on validation data then it might be enough to take its predictions and discontinue a category at a certain point. However, from a business perspective this might not be a good idea. The business might just want to use database lookups with data pushed to a dashboard where they can monitor categories and decide based on given conditions if they should discontinue.

```
15
16   Q5(nodeID) :-
```

Thank You