Data Scientist, Ops Research Test

Expectations

- Please use Python3 to complete this test. Take care to ensure that your code can be inspected and run easily. Feel free to use any libraries or tooling with which you're comfortable!
- Please provide all code, data, images, and text responses in a zip folder with the filename format *firstname-lastname.zip*.
    - You are welcome to collect your responses in a Jupyter notebook, or as a collection of scripts, text files, CSVs and images. Just make clear what files correspond to what section and question.

ETL

- You can find descriptions about the datasets to be used in this section in the **appendix** at the bottom of this test.

1. For every market and delivery window combination, find the number of orders placed and number of unique shoppers who made themselves available to fulfill an order.
2. Now, add two columns to the previous table in (1):
    a. the number of unique shoppers who abandoned their scheduled delivery window (hint: use the abandoned field)
    b. the number of unique shoppers who fulfilled an order in the delivery window
3. Save the resulting table from (2) as a CSV. This table should have six columns:
    a. Market
    b. Delivery Window
    c. # of Orders
    d. # of Unique Shoppers Available
    e. # of Unique Shoppers Who Abandoned
    f. # of Unique Shoppers Who Fulfilled

Visualization

1. Generate a graph or set of graphs that plots the # of orders, the # of unique shoppers available, and the # of unique shoppers who fulfilled from the ETL questions above against delivery windows. Make sure to visually distinguish each market's respective trends over time.

Free Response

Answer each question in no more words than what is listed before the prompt. There's no requirement to write any code to answer these questions.

1. (750 words max) We're interested in proactively identifying the amount of pay we need to attach to an order for it to be claimed and fulfilled by our shoppers, aka a "valuation model". We would like to build this model to understand how to better price orders to ensure they're efficiently cleared from the market. Assume you have a rich dataset of orders that includes information about:

   - When, where, and for what delivery window an order was placed
   - What was contained within the order
   - What shoppers the order was shown to, when it was shown to them, and how much pay we attached to that offer (in $)
   - Who ultimately claimed and fulfilled the order and when they claimed it

   Note that we can offer an order at multiple levels of pay at different times.

   a. How would you curate a dataset and design a model to tackle this problem? Be specific about the framing, approach, and techniques you might use and what features you think might be valuable to incorporate into your solution.

   b. How would you measure the success of your solution? What kinds of metrics would you construct to identify where the model performs well and where it struggles?

I would curate a dataset by talking to stakeholder(s) and making sure I have the right problem statement. Then I would query the database to grab relevant data.

The framed problem is:

For a given market, in a given neighborhood, on a given day, how do we maximize batch clearance by shoppers by presenting various pay at different times of the day?

For each market's neighborhood:

Nice-to-have additions to feature set:

- Traffic details
- When is the store busy?
- How long are lines usually?
- If order is accepted, will there be a subsequent batch to complete to maintain $15/hr wage?
- What is the total distance from shopper's home to store to customer and back home?
- Will there be a tip?

Target value: Price Attached to Order

Type of model: Linear Regression

<u>Exploratory Data Analysis</u>

Upon getting the initial dataset, I would do some exploratory data analysis to examine missing values and relationships between variables.

1. Numerical Features

I would look at all numerical features' distribution, including statistical characteristics (mean, median and mode). Distribution plots would help visualize data's distribution. Boxplots would reveal statistical characteristics and pinpoint outliers. From the various plots, I would take note of what stands out--plots with single values , skewness, any Normal distribution or lack thereof, etc.

2. Categorical Features

Countplots would help visualize within each feature, each distinct value's count. Here, I would take note of any categorical feature that consists of mainly one value, which would not be useful information.

3. Bivariate Features

Correlation Matrix is an effective tool to uncover linear relationships (correlation) between any two continuous features. I would single out any variables that have high correlation to one another by pinpointing correlation coefficients closest to 1. Variables that have high correlation to each other (not to target variable) are those with multicollinearity. Predictors with strong relationships to target would be determined by sorting correlation coefficients in descending order. Those at the top would have the strongest linear relationship to the target variable.

Last exploration would involve using scatter plots to identify any relationships between numerical features that the Correlation Matrix did not detect due to quadratic/exponential relationships. Scatterplots with polynomial regression lines plotted help define trends which in turn make outliers easily visible as well.

4. Feature Cleaning

To handle missing values/NaN's, would fill with 'None', 0 or mode based on what is realistic or trend in variable. Outliers would be removed by taking out anything outside of z-score with three standard deviations. An alternative method to using z-score would be to use Isolation Forests which detect anomalies.

5. Feature Engineering

For all rare values within each feature (<10), I would group them into 'Other' group as new value.

For categorical features, would plot them against target variable using boxplots. Median values within boxplots help highlight any within feature relationships. Category values that fall within same median value would be grouped together. Key quality features would be turned into numerical ones so closer categories have linear relationship.

Multicollinearity would be handled by merging similar independent variables into one and dropping similar ones from the dataset.

Predictor variables with strong relationship to target variable would be multiplied by similar numerical features so their influence is strengthened.

Simplifying features that have a range of values to 0/1 for presence or absence of a certain feature would be next. Also, can reduce distinct numerical values count within a feature to 4 time frames in the day for time feature in our dataset.

Transforming non-normal distributions (skewed) via boxcox transformation would help.

Last steps would be to drop features that would not be useful in predicting Pay Attached to Order and to one-hot encode remaining categorical features.

6. Modeling

I would try all of these:

- Ridge
- Lasso
- ElasticNet
- Support Vector Regression
- Gradient Boosting Regressor
- LightGBM Regressor
- XGBoost Regressor
- Hist Gradient Boosting Regressor

- Tweedie Regressor

Tuning models would be done with Optuna package.

Cross Validation would be performed to increase generalization of model. Based on test RMSE, would determine which model to move forward.

7. Ensemble Methods

And finally, I'd stack and blend models to get lowest RMSE. And for deployment, since Pay Attached to Order can be a range, would attach to each order point forecasts (mean) with 80%-95% prediction interval to offer multiple levels of pay at different times.

When framing the problem, some heuristics would be stated to create threshold values for success. RMSE would be used to measure success of solution or not and to identify where models perform well and where they struggle.

2. (500 words max) Something critical to our business is understanding what supply and demand might look like in the near future in order to plan for hiring and shopper activation activities. Let's say that you've collected the data from the ETL section for the period of a year:
    a. How would you approach forecasting our expected daily supply and demand at the market level?
    b. How would you incorporate information about the spatial hierarchy of supply and demand (i.e. neighborhoods roll up into a single market) and information about the "true" availability of supply (i.e. shoppers who signup but always abandon) into your forecasting method?
    c. How would you evaluate the validity and accuracy of your forecast?

My approach in forecasting expected daily supply/demand at market level:

1. Framing the problem:

This is a Time Series prediction problem for supply and demand in three different markets (2 different graphs with 3 markets per graph).

2. Gathering the necessary information:

This includes collecting the year's worth of statistical data and making sure it is enough to not cause a short time series (granularity at day level is sufficient). Sufficiency also depends on the number of model parameters to be estimated and the amount of randomness in the data. Also, there needs to be experts gathered who collect the data and use the forecasts.

Given that this is an on-demand business making real-time predictions, data source needs to be refreshed frequently.

3. Preliminary exploratory analysis

I would plot the as-is data and look for anything consistent, significant trends (growth), seasonality, evidence of business cycles, odd dips, outliers. Also box plots on variable(s) of interest would be good to see distribution of values, any outliers if present.

4. Clean data for modeling

Here I would deal with missing data by either getting rid of it, filling with 0 or mode based on percentage of missing values and trend in data. Outliers would be taken care of using z-score with 3 SD. Also, any relevant filtering to the data that needs to be done would be completed here. For example, "true" supply needs to be predicted on which means shoppers who did not abandon are the only shoppers included in supply forecast data.

5. Choosing Model to Fit Data and Make Predictions

Though various models can be used, for single series, will use Facebook's Prophet since it is easy and fast to implement. All that needs to be defined is whether or not there is seasonality to fit the model. And to make predictions, future dates need to be defined for forecasts. Predictions are output along with components and uncertainty intervals. Plots of forecast show trend, yearly seasonality and weekly seasonality.

6. Assessing Model's Performance

RMSE will be used to evaluate a model's performance. Lower the RMSE, better the model. When framing the problem, a measure of success is defined in order to compare model's RMSE to it.

Hierarchical Time Series

Usually, forecasting is done with single time series (one variable against dates), but there can be forecasts for hierarchical variables (cities within states within countries). In this situation, we need to complete hierarchical time series forecasts to capture spatial hierarchy of supply and demand (neighborhoods within cities/markets).

The problem is a 2-level hierarchy with total supply/demand at the top. The total is disaggregated into 2 series. The level below the top aggregate is the 3 different cities and at the bottom level are all the neighborhoods per city. Use Facebook Prophet's HTS package to perform all forecasts. Different time series forecasts at different levels are made by having one plot of total supply/demand forecast with cities' forecasts below it. Another plot would be all the various neighborhoods' forecasts in one plot.

3. (500 words max) Efficient supply allocation across our available delivery windows is key to optimizing how much the business spends on fulfilling orders. Too few shoppers in a window means that we lose out on deliveries, while too many means that we're wasting productive time that could be allocated elsewhere. Let's say you've collected the data from the ETL section for a period of a year:
    a. What additional information would you need to determine the cost of undersupply (too few shoppers) and the cost of oversupply (too many shoppers) in a given market x neighborhood x delivery window combination?
    b. Your team has produced a demand forecast model and has estimated the demand at every market x neighborhood x delivery window combination in the next week. Given you have the costs of under/oversupply on hand from (a), how would you approach determining the optimal number of shoppers we need to be available for each market x neighborhood x delivery window combination?

Additional information I would need: demand and cost of undersupply/oversupply. On top of this, other data that would be nice to have: traffic, weather, special events, whether or not a shopper works multiple gigs (this would decrease their availability for shifts).

With the demand forecast model and cost of not hitting supply right for supply-demand equilibrium, this is how I would approach determining the optimal number of shoppers needed to be available:

Define problem statement:

For a given market x neighborhood x delivery window combination with weekly granularity,

what are the optimal number of shoppers to minimize shopper idleness and to minimize lost deliveries?

This problem is hard due to variances in:

-demand forecast

-space-time density of demand

-shopper abilities

-cancellations

-shoppers’ mode of transportation

-weather

-traffic

Another thing to note is summary of costs: increased idleness means increased labor costs and loss in deliveries means decrease in revenue. Also, in the long run, loss in deliveries means losing potential repeat customers.

Heuristics need to be set up as methods for staffing. And then course correct on these over time. Some preliminary ones would be:

a. Forecast using previous week’s staffing numbers
b. Correct for next week based on previous week’s idleness and loss deliveries
c. Adjust for next week’s demand forecast changes

Caveat here is that dependency on previous week’s outcomes can result in a vicious feedback loop of settling at local minimum. It is more optimal to put shoppers in a store location with more ordering density around it.

In order to handle all variances from complex problem statement, Monte Carlo simulations would be initiated. Reason for this choice: no dependencies on prior week’s outcomes, able to model complex systems as interactions between random variables and able to perform stochastic optimization to deal with variances better.

There would be a lot of simulations of supply and demand. And from this, solve for set of staffing levels that minimize idleness, lost deliveries across all simulation runs.

Orders would be simulated by looking at how many orders there would be at a future date. Using a demand forecast’s estimated point value would not be sufficient due to variance. So solution would be to use distribution of demand forecast based on error distribution to determine number of orders needed to fulfill each simulation. Average from distribution would be the demand forecast.

To simulate shoppers, gamma distribution would be used since some shoppers are slow and others are fast. It would not be static in space.

This problem is essentially a Vehicle Routing Problem (fulfillment problem) with time windows. With this, the goal is to have an optimal set of routes. And from this, you can work backwards and derive how many shoppers are needed while minimizing under/over supply cost. Multiple simulations of this are done. Repeat process 100’s of times by repeatedly sampling and solving for staffing to account for variability. This is all done for every market x neighborhood x delivery window combination.

4. (500 words max) Shopping at Shipt has a steep learning curve, and we’re always interested in understanding how we can quantitatively evaluate the skills of our shoppers. There are a number of factors that qualify as skill, such as ability to

concurrently shop multiple orders, deep understanding of a particular store location's layout, or the sequence in which a shopper fulfills an order.

  a. What are two metrics that you can come up with that you believe are highly-correlated with shopper skill? How are these metrics calculated?
  b. What approach could you take to empirically test whether your metrics are useful, and what kind of data would you need to collect to do this?

Two metrics that I would come up with that I believe are highly-correlated with shopper skill are:

1) Accuracy of picked items including substituted items
   a) Calculated by: (picked items/listed items)*100%
2) Total time duration from start of shopping to delivery of grocery items to customer
   a) Calculated by measuring elapsed time in min

Approach I could take to empirically test whether my metrics are useful:

Look at shopper's job requirements as judgment of success and key results. Do metrics help measure whether or not key results are reached? Would measurements from defined metrics help separate non-skilled shopper from skilled shopper?

Would answer these questions by performing Hypothesis test:

Null Hypothesis: Metrics do not help distinguish non-skilled shoppers from skilled shoppers.

Alternative Hypothesis: Metrics do help distinguish non-skilled shoppers from skilled shoppers.

Kind of data I would need to collect to do this:

1. Customer's shopping list (post-substitutions), shopper's checkout items
2. Duration of time between start of shopping and delivery of grocery items to customer

Classic A/B Testing: to confirm hypothesis that chosen key measures are different in treated group (those that received metrics) to evaluate treatment's effects (helped in distinguishing non-skiller shopper from skilled shopper)

1. Make sure there are randomized groups for control group and treatment group
2. When determining sample size:
   a. Need to make sure it is large enough to yield high statistical power
3. Data Pre-Processing:
   a. ratio between control and treatment groups not significantly different
   b. Check Metric: What type? This determines hypothesis testing process type
      i. Continuous
      ii. Ratio
      iii. Proportional

In this situation, it is continuous for both metrics.

4. Outlier Detection/Variance Reduction/Pre-Experiment Bias
    a. Need to be sensitive to these so that they do not pollute dataset
5. P-Value Calculation (Significance)
6. Sample Size can come in different sizes:
    a. Skewed
    b. Small
    c. Large

    Based on these, there are different tests that are run:

        1. T-test
        2. Chi-Square Test

7. Lift

    a. see if there is a significant difference in treatment group

8. Power Calculation performed in order to see the level of confidence in analysis

Experimental outcomes that constitute success/failure will be based on alpha commonly set at 0.05 as the cutoff for significance. If the p-value is <0.05, we reject the null hypothesis that there is no difference between the means from each test group and conclude that a significant difference does exist (my metrics do help distinguish non-skilled shopper from skilled shopper).

However, what would negatively impact the validity of the experiment would be "network effect". The success of the new feature could simply be due to other reasons outside of the treatment group. Validity of A/B tests rest on treatment only affecting behavior of treated users, which is not always the case.

In order to remedy this, researcher performing experiment could assign treatments randomly based on clusters of users rather than individual users. Network effect is then worked against in experiment.

Appendix

There are three datasets provided with this test. The following is a listing of the fields contained in these tables and short descriptions about what they are.

Table	Field	Description	Type
delivery_windows.csv	delivery_window_id	Unique id for a delivery window, an hour-long window on a particular day in a particular market	Integer
delivery_windows.csv	starts_at	Start of the hour for the delivery window	Datetime
delivery_windows.csv	market_name	Name of the market	String
delivery_windows.csv	neighborhood_name	Name of the neighborhood	String
shopper_schedule.csv	shopper_id	Unique id for a shopper	Integer
shopper_schedule.csv	delivery_window_id	Delivery window that the shopper made themselves available to fulfill orders	Integer
shopper_schedule.csv	abandoned	Shopper signaled that they were not available to shop during this time	Boolean
orders.csv	order_id	Unique id for an order	Integer
orders.csv	delivery_window_id	Unique id for a delivery window	Integer

orders.csv	shopper_id	Unique id of shopper who fulfilled this order	Integer