Data Modeling Exercise

NIQ

Preface

This take home assignment is designed to assess the candidate's proficiency in executing tasks that are commonly performed in data-related roles at NIQ. The following exercise is split as follows:

0. Introduction
    - Helps to get familiar with the main dataset
1. Basic Data Preparation
    - Tests standard knowledge of data queries in python
2. Data Description
    - Tests ability to visualise and interpret patterns in data
3. Data Wrangling
    - Tests data manipulation skills in python
4. Data Modeling
    - Tests knowledge of simple modeling techniques
5. Further Data Modeling
    - Tests knowledge of more involved modeling techniques (please read instructions under this sections)

General advice on completing this data exercise

- Each correct answer is rewarded with a point, while unanswered questions and wrong answers do not affect the score (aka Positive Marking). Try to complete all tasks to the best of your ability
- Candidates are encouraged to focus solely on the tasks outlined in the exercise. Points will be awarded only for correct answers to the exercise questions

0 Introduction

Start by loading the data (**df1.csv only**) into memory. This is a standard store level data-set for a group of Stock Keeping Units (SKUs). In the data we should have the following columns:

- ITEM_ID - Unique code associated with each item
- ITEM_DESCRIPTION - Item name with its associated number
- CHANNEL - Market that is covered by the database
- STORE_ID - Unique store id. Id values do not have any meaning
- PERIOD_ID - Unique week id with meaningful values. The lowest value represent the first week in the data and the largest value represents the last. Each subsequent iteration from the first week adds a single week
- SALES_UNITS_EXPANDED - Number of units sold
- SALES_VOLUME_EXPANDED - Volume sold
- SALES_VALUE_EXPANDED - Value of units sold

ITEM_ID	ITEM_DESCRIPTION	STORE_ID	PERIOD_ID	SALES_UNITS_EXPANDED
1	001_Item	31905	70	1
1	001_Item	31905	71	3
1	001_Item	31905	72	1
1	001_Item	31905	73	4
1	001_Item	31905	77	1

Each row is interpreted as data for a single SKU at a given time period at a given store. So for example, the first row tells us that 1 unit of 001_Item was sold at store 31905 and week 70.

©2024 NIQ. All rights reserved.

1 Basic Data Preparation

The following set of tasks will test familiarity with basic data manipulation techniques.

1. Generate a variable called ACTUAL_PRICE
   This will be the associated price per unit for a particular SKU at a given store week. Use the available columns to obtain this value

2. Write a pseudo code (aka algorithmic logic) for generating REGULAR_PRICE and PROMO variables
   - REGULAR_PRICE - price per unit in absence of any promo
   - PROMO - a binary variable that indicates for each observation whether price promo was carried out
   Context: When we work with store week sales price data we want to know at the most granular level when price promo is executed. We are given no indicators by stores when they are offering discounts and our job is to figure it out by ourselves using this data.

2 Data Description

In this section we want to get familiar with the data set

1. Draw a histogram of SALES_UNITS_EXPANDED for **each item**

   (a) What distribution does it follow? It is the same for all items?

   (b) Does the answer change if we applied a log transformation?

2. Draw SALES_UNITS_EXPANDED and ACTUAL_PRICE on a single graph with PERIOD_ID on the x-axis for **each item**

   (a) Do you notice any seasonal patterns? Is it similar for all items?

   (b) Can we deduce that price reductions lead to sales increase from this graph? Why/Why not?

   Now plot a second graph with SALES_UNITS_EXPANDED and number of shops with PERIOD_ID on the x-axis

   (c) Does this support your conclusion in (b)?

3 Data Wrangling

In this section we want to get a feel for working with sales price data

Load the additional (**df2.csv**) data set
To understand REGULAR_PRICE and PROMO review the definition in 1.3
PROMOTION_PRICE_INDEX - is a ratio of ACTUAL_PRICE and REGULAR_PRICE

1. Generate a binary variable called PROMO_SHOP
   PROMO_SHOP is defined as stores where the **longest consecutive sequence** of PROMO=1 covers at least 75% of the total entries, indicating a sustained period of promotional activity.
   For example, for a single item-shop PROMO sequence [1,1,1,0] PROMO_SHOP would be 1. But if it was [1,1,0,1] then PROMO_SHOP would be 0

2. Produce a data summary **for each item** for the following columns **using pandas groupby**:

   (a) Sum for SALES_UNITS_EXPANDED (Total Sales)

   (b) Sum for SALES_UNITS_EXPANDED where PROMO = 1 (Promo Sales)

   (c) Weighted average value for ACTUAL_PRICE for all shops (Weighted Price)

   (d) Weighted average value for REGULAR_PRICE for shops where PROMO_SHOP = 0 (Weighted Regular Price)

   - Choice of column for weights is up to the candidate

3. Comment on the values in 3.2

   (a) Which items appear to be similar to each other in terms of weighted regular price? What about weighted price?

   (b) Which item has the highest sales? Does it also have the highest share of promo sales?

©2024 NIQ. All rights reserved.

4 Data Modeling

In this section we will try to estimate the impact of our independent variables on our dependent variable

Assume: Data generating process (DGP) for each item is given by

$$y_{it} = x_{it}^{\beta} * e^{\epsilon_{it}} \tag{1}$$

- y_{it} is a scalar value of dependent (aka target) variable in group i at time t
- x_{it} is vector of independent variables in group i at time t with β being a vector of associated coefficients
- ϵ_{it} is a scalar value of idiosyncratic error in group i at time t

Let x_{it} be REGULAR_PRICE, PROMO and PROMOTION_PRICE_INDEX
y_{it} be SALES_UNITS_EXPANDED
i be STORE_ID and t be PERIOD_ID

1. List necessary and sufficient assumption to get a consistent and asymptotically normal estimate of the marginal impact of price parameters on sales using Ordinary Least Squares (OLS)?
    - Do we require normality of errors to obtain consistent estimate?
    - How does violating spherical errors assumption impact consistency of the estimate?
2. Estimate β for **each item**
    - Hint: Apply an appropriate monotone transformation to convert this model to something that can easily be estimated by OLS
    - Hint: Think about whether we need to apply it to all variables (or only some)

    (a) Interpret coefficient estimates. Which sign would we expect? What values did we get?
    (b) Perform t-test and F-test. What can we infer from this? Are these test values accurate?
        - Hint: Recall how t and F statistics are calculated
    (c) Do you think this multiplicative specification is more realistic compared to standard linear model? Explain

Now we want to see how our coefficients would be different in different time periods.

Change the DGP in (1) to account for the following new assumptions:

- PROMO and PROMOTION_PRICE_INDEX coefficients should be different by season
- REGULAR_PRICE should not be impacted by seasonality

Seasons are defined as follows:

- Season 1 starts at PERIOD_ID = 10 and ends at PERIOD_ID = 41
- Season 2 starts at PERIOD_ID = 42 and ends at PERIOD_ID = 73
- Season 3 starts at PERIOD_ID = 74 and ends at PERIOD_ID = 104

3. Estimate β for **each item** given our new assumptions
    (a) Interpret these new coefficients? Compare them to estimates from 3.2
    (b) Perform t-test and F-test. What can we infer from this?
    (c) Do you think this seasonality assumption is accurate? Why/Why not?

©2024 NIQ. All rights reserved.

5 Further Data Modeling

This section continues the estimation process from section 4

Information for section 5 only:

- Each section is meant to be a standalone task (i.e. no need to do 5.1 to proceed to 5.2 for example)
- It is acceptable to assume that **all coefficients are the same across seasons** for simplicity (i.e. ignore 4.3 specification)
- **Candidate is not required to complete this whole section.** Approach to these problems will be discussed during the interview process
- **It is recommended to attempt one subsection in detail** and write down notes for others

5.1 Basic Feature Engineering

Now, we want to test the impact of each item on each other. To do this, in addition to own independent variables from a given item, we need to add to our x_{it} values from competitor items. So for example, a model for 001_Item would include its own REGULAR_PRICE, PROMO and PROMOTION_PRICE_INDEX and then same values for each competitor. Also, we want to transform REGULAR_PRICE for competitors only into ratios such that

$$RPR_{1j} = \frac{\text{001\_Item REGULAR\_PRICE}}{\text{00j\_Item REGULAR\_PRICE } \forall j}$$

1. Generate RPR_{ij} for for **item of choice**
2. Generate a variable that would track the presence of each competitor in each store-week
3. Estimate β updated model for **item of choice**. How would you interpret these values? What do they imply?

- Hint: Try to think of a way to fill NA values for competitor items store-week with some value

5.2 Residualization

Let $e^{\epsilon_{it}} = e^{\alpha_i + \omega_t + \upsilon_{it}}$ and use this definition from now on

- α_i is a shop specific effect which varies by i (e.g. turnover) with $E(\alpha_i | x_{it}) = 0 \ \forall i, t$
- ω_t is a week specific effect which varies by t (e.g. seasonality) with $E(\omega_t | x_{it}) = 0 \ \forall i, t$
- υ_{it} is the remaining idiosyncratic term with $E(\upsilon_{it} | x_{it}) = 0 \ \forall i, t$

1. How does this change the estimation procedure in 3? What adjustments, if any, are needed to ensure the consistency of coefficients? Explain
    - Hint: Recall assumptions from 3.1 and think how (or whether) these conditions impact them?

Now assume $E(\alpha_i | x_{it}) \neq 0$ and $E(\omega_t | x_{it}) \neq 0$

2. Estimate β under these new assumptions for **item of choice**. How does the estimate change from the before? What does it say about our data?
    - Hint: Try to think what does say seasonality impacting price variables imply? What about shop size and price variables? How can we account for that?

5.3 Custom Optimisation

Using any version of Least Squares (LS) estimator on a linear additive (or multiplicative) model we are relying on a closed form solution to the following expression:

$$\hat{b} = \underset{b \in B}{\arg\min} \sum_{i=1}^{N} \sum_{t=1}^{T} (q_{it} - p_{it} b)^2 \tag{2}$$

Since we are not placing any restrictions on our estimation process it can result in coefficient result that do not make business sense. For example, coefficient for REGULAR_PRICE being significantly higher than PROMOTION_PRICE_INDEX which indicates that customers are impacted by long term price changes way more than they would by temporary price reductions (which is unlikely to be true).
One way to solve this is to restrict the parameter space according to our prior beliefs and hence turn the problem into a constrained optimisation.

©2024 NIQ. All rights reserved.

1. After applying monotone transformation to (1), define a custom function (with unknown variable β) that would calculate a squared loss for our $x_{it}\beta$ and y_{it}
2. Add a constraint function that restricts our coefficients as follows: $|\beta_{PROMOTION\_PRICE\_INDEX}| > |\beta_{REGULAR\_PRICE}|$ along with an upper bound of 0 for both coefficients
3. Estimate coefficients for **item (where PROMOTION_PRICE_INDEX coefficient was lower (in absolute terms) than REGULAR_PRICE)** with loss function defined in 1 and constraint defined in 2. How would the result be different if we adjusted coefficients manually post estimation? Explain

- Hint: Recall that this is a simple quadratic programming problem (so it is possible to set up a Jacobian function)

©2024 NIQ. All rights reserved.