Timed Work Assessment

Overview / Purpose

Advertising campaigns provide a boost to sales through visibility. This comes at a cost, usually per click, whenever a user interacts with our ad. Each advertising campaign consists of a set of keywords. Associated with a keyword are the fields: **bid**, **state**, **impressions**, **clicks**, **cost**, **sales**, and advertising cost of sales (**acos**). Each campaign is associated with a product which has a **breakeven acos** and a **breakeven cost**. Given these metrics and thresholds, our goal is to adjust keyword bids based on observed performance to find an optimal bid. ==**Your task is to generate an output file (Excel, .xlsx) with bid and state proposals.**==

==**Goals (time goals, general goals, etc.)**==
- ==Time Goal: 2 Hours==
- ==Commenting Goal: Clear commenting on functions created and processing flows==
- ==Review Goal: Output file should be easily reviewed for accurate calculations and==
- ==Feedback Goal: Provide comments, thoughts, or concerns about output generated==

Requirements
- Python
- Pandas

Details:
- Overview of Components / Inputs
    - ☑ Keyword Performance Data (1 of 3 input files)
        - ☐ Each row represents an observation of a keyword's performance for a given date.
        - ☐ Performance-related Fields to note are: **impressions**, **clicks**, **cost**, **sales**, and **units sold**.
        - ☐ Other fields to note are:
            - The keyword_id field can be used to uniquely identify a keyword
            - short_id field indicates the product associated with a row's performance
    - ☑ Keyword Bid and State Data (2 of 3 input files)
        - ☐ Each row represents a keywor_id and its bid and state values.
    - ☐ Product Database (3 of 3 input files)
        - ☐ This file contains the breakeven ACoS and breakeven cost values for each product. Each row in the keyword performance dataset has a "short_id" field. The short_id represents a product and you can use it to map the breakeven values for use in bid optimization rules.

- General/Summary of Processing Flow
    - ☑ 1. Read keyword performance data (wa2_keyword_performance.xlsx)
        - ☐ This file contains one record per keyword_id per day.
    - ☐ 2. Use the "date" field to determine the most recent date observed.This will be the ending value for your date range, inclusive.
    - ☑ 3. Determine the date 30 days before the ending date range value. This will be the starting value for your date range, inclusive.
    - ☑ 4. Aggregate the metrics across desired time period and calculate the ACoS for each keyword_id across the date range.

- ☐ This should produce a DataFrame with one record per keyword_id
- ☐ It would be helpful to keep unaggregated values for match_type, campaign_id, campaign_name, short_id, top_search_multiplier
- ☑ 5. Read the keyword bid file (wa2_keyword_bid.xlsx) and map bid onto the aggregated performance table via keyword_id.
- ☑ 6. Given bid and top_search_multiplier, calculate effective_bid
  - ☐ effective_bid = bid * top_search_multiplier
- ☑ 7. Apply bid optimization rules (see Processing section below) to keywords based on the aggregated metrics across the desired time period, proposing a new effective_bid value or state as necessary.
- ☑ 8. Determine Final Proposed Bid and Final Proposed State based on the applied optimization rule
  - ☐ For Final Proposed Bid: Be sure to calculate the (raw) bid from the Final Proposed Effective Bid (see effective bid section in the table above) as bid = effective_bid/top_search_multiplier
- ☑ 9. Note the bid optimization rule applied/type of change (ex: bid increase, bid decrease, pause)
- ☑ 10. Generate output file

- **Bid Optimization Rule Details**
  - ☑ **Bid Optimization Rules**
    - ☐ Rule 1
      - If acos > (breakeven acos + 0.10), then decrease effective bid by 25%
        - ○ Ex: Breakeven ACoS = 0.20,
        - ○ Breakeven ACoS + 0.10 = 0.30
    - ☐ Rule 2
      - If acos < (breakeven acos - 0.05) AND at least 2 units ordered, then increase effective bid by 15%
        - ○ Ex: Breakeven ACoS = 0.20,
        - ○ Breakeven ACoS - 0.05 = 0.15
    - ☐ Rule 3
      - If there are no sales and spend is greater than 1.25 * breakeven cost, then decrease effective bid 25%
    - ☐ Rule 4
      - If acos > 2.75 * breakeven acos, set Final Proposed State to paused
    - ☐ Rule 5
      - If there are no sales and cost > 2.5 * breakeven cost, then set Final Proposed State to paused
  - ☑ Bid Floors and Bid Roofs
    - ☐ For keywords belonging in Top Keywords campaigns (campaign_name includes 'Top Keywords'), the effective bid should not exceed 10.
    - ☐ For all other keywords, the effective bid should not exceed 8.
    - ☐ All keywords should have raw bids no lower than 0.02.
- **Output**
  - ☑ Excel file, with each row corresponding to a given keyword id
  - ☑ Columns
    - ☐ keyword_id

- ☐ keyword_text
- ☐ campaign_name
- ☐ short_id
- ☐ Current bid
- ☐ top_search_multiplier
- ☐ Effective Bid
- ☐ Impressions (aggregated over desired time period)
- ☐ Clicks (aggregated over desired time period)
- ☐ Cost (aggregated over desired time period)
- ☐ Sales (aggregated over desired time period)
- ☐ ACoS (aggregated over desired time period)
- ☐ Units Sold (aggregated over desired time period)
- ☐ Conversions (aggregated over desired time period)
- ☐ Breakeven ACoS
- ☐ Breakeven ACoS+10
- ☐ Breakeven ACoS-5
- ☐ Breakeven ACoSx2.75
- ☐ Breakeven Cost
- ☐ Breakeven Costx1.25
- ☐ Breakeven Costx2.5
- ☐ Bid Optimization Rule(s) Applied (can use the Rule # and/or the rule description)
- ☐ Final Proposed Effective Bid
- ☐ Final Proposed (Raw) Bid
- ☐ Final Proposed State
- Potential Pitfalls
  - ☑ Check that calculated performance acos is in the same format as breakeven ACoS from the product database
  - ☑ The product database is manually updated (in practice) and there may be missing fields. If a value is missing, assume breakeven ACoS = 20% (0.20) and breakeven cost = 7.50
  - ☐ Although an .xlsx output is ideal, if it proves troublesome, a CSV file is an alternative
- Testing / Reviewing / Logging
  - ☑ Indicate in comments, output, or a feedback summary regarding any methods used to verify output accuracy
  - ☑ Check off features in this document which have been implemented (clicking the bullet point box will check off the bullet point)
  - ☑ Note: The columns requested for the output generally help in manually reviewing output for desired processing. Feel free to include any other columns that you feel may help

Supplemental resources
- Installing Excel packages for Pandas
- Optimizing Your PPC
  - Beginner's Guide / Introduction

Definitions

bid (raw	The amount we pay for an advertisement spot (>= 0.02)

bid)	
Top of search multiplier	Advertisement spots have different placements. They can be at the top of the search results or scattered throughout search results. This value scales a keyword's bid to provide a top-of-search-specific bid. This value can be between 0 and 10.
Effective Bid	Effective Bid = Bid * Top of search multiplier Ex1: Bid: 2, Top of search multiplier: 1 Effective Bid = 2 = 2 * 1 Ex2: Bid: 0.95, Top of search multiplier: 10 Effective Bid = 9.5 = 0.95 * 10 Similarly, given an effective bid we can calculate the (raw) bid. Bid = (Effective bid)/(Top of search multiplier)
state	Whether the keyword is active (enabled, paused)
impressions	The number of times an ad has been viewed
clicks	The number of times an ad has been clicked
cost	The total cost of the ad across all clicks
sales	The total sales generated from clicks to the ad
conversions	The number of transactions, where one transaction can include multiple items
units ordered	The number of units sold through transactions, where one transaction can have multiple units.
acos	Advertising Cost of Sales (ACoS) acos = cost/sales The ratio of spend to sales.
breakeven acos	Each product will have a static breakeven acos (that doesn't vary over time). The breakeven acos is the threshold ratio of cost to sales where an observed ACoS below the breakeven acos would generate a profit. An observed acos above the breakeven acos results in a loss (net zero if observed acos = breakeven acos)
breakeven cost	Each product will have a static breakeven cost (that doesn't vary over time), where generating 1 sale at a cost below the breakeven cost results in a profit. Generating 1 sale at a cost above the breakeven cost generates a loss (net zero if cost to generate the 1 sale is the same as break evencost.