TECHNICAL DOCUMENTATION

Survey Analysis Code

Data Cleaning, Two-Stage PLS-SEM, Clustering, and Decision-Tree Profiling

A guide to Source_Code1.ipynb and Source_Code2.ipynb

Documentation date: August 23, 2026

1. Purpose and scope

These two notebooks implement a linked survey-analysis workflow. The first notebook cleans and preprocesses item-level survey responses, describes the data, visualizes construct correlations, and estimates a first-order partial least squares structural equation model (PLS-SEM). The second notebook re-estimates the first-order scores, constructs a higher-order serial model, evaluates model quality, segments respondents, and profiles the segments with a decision tree.

> **Model pathway:** Expertise -> Parasocial -> Relationship Quality -> Behavioral Response. Relationship Quality is formed from Trust and Loyalty; Behavioral Response is formed from Purchase Intention and Purchase Behavior.

1.1 Pipeline at a glance

1. Load food2.csv and inspect 34 survey items plus seven demographic/use variables.
2. Remove duplicates and one manually excluded record, convert fields to numeric values, and impute missing data.
3. Create construct means, descriptive statistics, correlations, and the first-order PLS-SEM measurement model.
4. Use the first-order latent-variable scores as indicators in a second-stage hierarchical PLS-SEM model.
5. Cluster higher-order scores into engagement groups and explain cluster membership using demographic and usage variables.

1.2 Notebook responsibilities

Notebook	Primary responsibility	Key persisted output
Source_Code1	Cleaning, imputation, construct scoring, descriptive analysis, correlations, and Stage 1 PLS-SEM	output.csv; correlation_heatmap.png; figure.eps
Source_Code2	Stage 2 hierarchical PLS-SEM, bootstrapping, model diagnostics, clustering, and decision-tree profiling	Decision-tree PNG/PDF files; figures displayed in notebook

2. Environment and data contract

2.1 Required Python packages

Package	Role
pandas	CSV loading, tabular cleaning, imputation, aggregation, and result tables
numpy	Array operations, correlation masking, and reliability calculations
scipy	Pearson correlation coefficients and p-values
matplotlib / seaborn	Correlation heatmap, cluster-selection charts, tree plots, and feature-importance plot
openpls-engine	PLS-PM / PLS-SEM estimation, bootstrapping, and model-quality statistics
scikit-learn	Standardization, K-means clustering, cluster-validity indices, and decision tree

```
pip install openpls-engine
# pandas, numpy, scipy, matplotlib, seaborn, and scikit-learn are also required
```

2.2 Input schema

The raw file is expected to be named food2.csv and to contain 41 columns: Q1-Q34 plus Age, Grade, Gender, Level, Place, Instagram, and Page. Survey items are treated as numeric Likert-type values after conversion. Demographic/use columns appear to be numerically coded except Age.

Field group	Columns	Expected use
Survey items	Q1-Q34	Measurement indicators for six first-order constructs
Age	Age	Continuous demographic predictor; median-imputed
Coded attributes	Grade, Gender, Level, Place	Descriptive/profile variables; some are used as decision-tree predictors
Platform behavior	Instagram, Page	Daily Instagram use and food-blogger browsing frequency/time codes

> **Execution order:** Run Source_Code1 first because Source_Code2 loads output.csv. The Stage 1 cells in Source_Code2 are then re-run to generate the latent scores used by the Stage 2 model.

3. Source_Code1: cleaning and first-order modeling

3.1 Load and initial profile

The notebook loads food2.csv into a pandas DataFrame. The attached run begins with 377 rows and 41 columns. It isolates Q1-Q34 as the response-quality subset dq while retaining demographic and usage columns in the full DataFrame.

3.2 Response-quality checks and exclusions

6. Counts literal single-space values in Q1-Q34 for each respondent and lists rows with at least one such value.
7. Computes the largest within-row response proportion and flags respondents for whom at least 90% of non-missing answers are identical. No rows are flagged in the attached run.
8. Counts the number of unique answers per row and flags single-pattern responders. No rows are flagged in the attached run.
9. Detects 127 exact duplicate rows and retains the first occurrence of each duplicate.
10. Deletes row index 91 because it is a reclassified response. After duplicate removal and this deletion, 249 cases remain.

3.3 Numeric conversion and missing-value treatment

All columns are converted with pandas.to_numeric(errors='coerce'), so nonnumeric entries become NaN. Missing survey items are then imputed according to construct membership:

11. Trust (Q6, Q7): each missing item is replaced with that item's sample mean.
12. All other constructs: a missing item is replaced with the respondent's mean across the available items in that same construct.
13. Age: missing values are replaced with the sample median.
14. Gender and Place: missing values are replaced with the modal code.

The cleaned 41-column dataset is saved as output.csv before the six construct-mean columns are added to the in-memory DataFrame.

3.4 First-order construct map

Construct	Indicators	Count
Expertise	Q1, Q2, Q4, Q5	4
Trust	Q6, Q7	2

Construct	Indicators	Count
Parasocial	Q3, Q8-Q15	9
Loyalty	Q16-Q24	9
Purchase Intention	Q25-Q28, Q30, Q31	6
Purchase Behavior	Q29, Q32-Q34	4

For descriptive analysis, each construct score is the row mean of its indicators. These mean scores are convenient summaries; the later PLS-SEM models separately estimate latent-variable scores from the item blocks.

3.5 Descriptive statistics and correlations

The notebook reports count, mean, standard deviation, minimum, maximum, skewness, and kurtosis for the six construct means. It then calculates all pairwise Pearson correlations and p-values. A lower-triangular heatmap is saved as correlation_heatmap.png at 300 dpi and again as figure.eps with reader-friendly labels.

Construct	Mean	SD	Min	Max	Skew	Kurtosis
Expertise	3.860	0.801	1.750	5.000	-0.586	-0.361
Trust	3.116	0.986	1.000	5.000	-0.179	-0.924
Parasocial	3.294	0.729	1.444	5.000	-0.268	-0.561
Loyalty	3.439	0.885	1.000	5.000	-0.698	-0.459
Purchase Intention	3.414	0.730	1.000	5.000	-0.497	0.170
Purchase Behavior	3.187	0.821	1.000	5.000	-0.091	-0.354

The strongest reported correlation is between Parasocial and Loyalty (r = 0.841), closely followed by Purchase Intention and Purchase Behavior (r = 0.840). The code calculates p-values, although its formatted correlation table does not add significance symbols because every p-value branch uses the same numeric format.

3.6 Stage 1 PLS-SEM

The Stage 1 model uses reflective Mode A measurement blocks with path weighting and standardized data. Its structural specification is:

15. Expertise -> Parasocial
16. Expertise -> Trust

17. Parasocial -> Trust
18. Trust -> Loyalty
19. Loyalty -> Purchase Intention
20. Purchase Intention -> Purchase Behavior

The Stage 1 structural model was specified as a theoretically grounded nomological network to facilitate the estimation of the lower-order construct scores required for the subsequent higher-order analysis. The sequence reflects the progression through which perceived influencer expertise contributes to parasocial relationships and trust, relational trust strengthens loyalty, and loyalty is translated into purchase intention and subsequent purchase behavior. This specification is consistent with prior influencer-marketing research demonstrating relationships between expertise, parasocial relationships, trust, loyalty, and purchase-related responses, as well as with the Theory of Planned Behavior, which identifies behavioral intention as a proximal antecedent of actual behavior. Importantly, the Stage 1 structural relationships were used primarily to embed the lower-order constructs within a meaningful nomological network and obtain their latent variable scores; the substantive higher-order relationships were evaluated separately in Stage 2.

> **Stage 1 versus Stage 2:** Stage 1 is a theoretically meaningful first-order network used for measurement assessment and latent-score estimation. Stage 2 is the higher-order model used for final structural hypothesis testing.

The notebook extracts latent-variable scores for all six constructs, reports Cronbach's alpha and Dillon-Goldstein's rho, and calculates composite reliability (CR) and average variance extracted (AVE) from the outer loadings.

Construct	Cronbach alpha	DG rho	CR	AVE
Expertise	0.847	0.897	0.897	0.685
Trust	0.801	0.909	0.909	0.833
Parasocial	0.876	0.902	0.902	0.514
Loyalty	0.928	0.940	0.940	0.636
Purchase Intention	0.800	0.860	0.859	0.515
Purchase Behavior	0.806	0.874	0.873	0.634

4. Source_Code2: higher-order modeling and segmentation

4.1 Recreate Stage 1 scores

Source_Code2 loads output.csv, verifies that no values are missing, installs openpls-engine in the Colab environment, and repeats the same first-order PLS-SEM specification. The six Stage 1 latent-variable scores are copied to stage2_df.

4.2 Stage 2 hierarchical model

The second-stage model is a serial path model built from Stage 1 scores. Expertise and Parasocial are represented by their single Stage 1 scores. Relationship Quality is measured by Trust and Loyalty scores. Behavioral Response is measured by Purchase Intention and Purchase Behavior scores.

Stage 2 construct	Indicators from Stage 1	Structural role
Expertise	Expertise score	Exogenous predictor
Parasocial	Parasocial score	First mediator
Relationship Quality	Trust score; Loyalty score	Second mediator
Behavioral Response	Purchase Intention score; Purchase Behavior score	Final outcome

4.3 Stage 2 estimates

Path	Coefficient	Bootstrap mean	SE	95% CI	t
Expertise -> Parasocial	0.744	0.743	0.035	[0.666, 0.803]	21.06
Parasocial -> Relationship Quality	0.879	0.878	0.013	[0.850, 0.903]	66.04
Relationship Quality -> Behavioral Response	0.750	0.751	0.029	[0.690, 0.805]	25.85

The model is re-estimated with 5,000 bootstrap iterations. The reported total serial effect from Expertise to Behavioral Response is 0.490 (bootstrap mean 0.490, SE 0.033, 95% percentile interval [0.422, 0.554], t = 14.83). This effect follows the complete Expertise -> Parasocial -> Relationship Quality -> Behavioral Response chain.

4.4 Measurement and structural diagnostics

Diagnostic	Result in attached run	Interpretation within the code

Diagnostic	Result in attached run	Interpretation within the code
R-squared	Parasocial 0.553; Relationship Quality 0.772; Behavioral Response 0.563	Explained variance for endogenous constructs
Q-squared	0.547; 0.663; 0.504	Positive predictive-relevance values
f-squared	1.238; 3.383; 1.289	Each active structural path is labeled large
Indicator VIF	2.199 for Trust/Loyalty; 3.426 for Intention/Behavior	Measurement-level collinearity check
SRMR	0.0664	Reported saturated-model fit index
HTMT	0.8398 for Relationship Quality vs. Behavioral Response	Only pair available between two multi-indicator Stage 2 constructs

4.5 Clustering of Stage 2 scores

The four Stage 2 scores are standardized with StandardScaler. K-means solutions from k = 2 through k = 10 are compared using within-cluster sum of squares (WCSS), silhouette score, Calinski-Harabasz index, and Davies-Bouldin index. In the attached run, k = 2 has the highest silhouette score (0.489), highest Calinski-Harabasz index (364.942), and lowest Davies-Bouldin index (0.751), so the notebook fits a two-cluster solution.

Cluster	Label	n	Expertise	Parasocial	Relationship Quality	Behavioral Response
0	Low Engagement	111	-0.822	-0.902	-0.947	-0.762
1	High Engagement	138	0.661	0.725	0.761	0.613

The cluster-profile values are means of standardized latent scores. Positive values indicate above-average engagement on the modeled constructs; negative values indicate below-average engagement.

4.6 Decision-tree profiling

Cluster membership is appended to the cleaned respondent data. A DecisionTreeClassifier is trained with maximum depth 4, minimum leaf size 10, and random_state = 42. Predictors are Age, Grade, Gender, Level, Place, Instagram, and Page. Publication-sized tree images are produced in PNG and PDF formats.

Predictor	Feature importance	Rank
Page	0.642	1
Grade	0.170	2

Predictor	Feature importance	Rank
Instagram	0.065	3
Place	0.064	4
Level	0.034	5
Age	0.024	6
Gender	0.000	7

5. Mathematical foundations

This section states the mathematics implemented or represented by the notebook operations. Subscript i denotes a respondent, j denotes an observed item, c denotes a construct, n is the number of respondents, and m_c is the number of indicators assigned to construct c. The openpls-engine package controls numerical normalization and convergence details; the equations below describe the corresponding estimands and algorithmic logic.

5.1 Missing values and construct scores

For Trust, a missing item is replaced by the observed sample mean for that item. If O_j is the set of respondents with an observed value for item j, the replacement is:

$$x^*_{ij} = \frac{\sum_{r \in O_j} x_{rj}}{|O_j|}$$

For the other constructs, the code uses the respondent's available-item mean within that construct. If O_ic is respondent i's observed indicator set for construct c:

$$x^*_{ij} = \frac{\sum_{k \in O_{ic}} x_{ik}}{|O_{ic}|}, \text{for} j \in J_c$$

Age uses the sample median, while Gender and Place use the modal category. After imputation, the descriptive construct score is the arithmetic mean of the construct's m_c indicators:

$$\text{score}_{ic} = \frac{\sum_{j \in J_c} x_{ij}}{m_c}$$

This equal-weight mean is used for descriptive statistics and correlations. It is distinct from the PLS latent score, which uses estimated outer weights.

5.2 Descriptive moments and Pearson correlation

For a construct score x, the sample mean and standard deviation are:

$$\bar{x} = \frac{\sum_i x_i}{n} \text{and} s_x = \sqrt{\frac{\sum_i (x_i - \bar{x})^2}{n-1}}$$

The notebook's skewness and kurtosis summarize distribution shape. In moment notation, with m_r = (1/n) sum_i (x_i - x_bar)^r:

$$\text{skewness} = \frac{m_3}{{m_2}^{\frac{3}{2}}} \text{andexcess kurtosis} = \frac{m_4}{{m_2}^2} - 3$$

pandas applies finite-sample bias corrections to these displayed estimates. Pearson's correlation between constructs x and y is:

$$r_{xy} = \frac{\sum_i (x_i - \bar{x})\,(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

The associated test evaluates H0: rho_xy = 0. With n observations, the usual test statistic is t = r sqrt[(n - 2)/(1 - r^2)] with n - 2 degrees of freedom.

5.3 Reliability and convergent validity

Cronbach's alpha for a k-item block compares the sum of item variances with the variance of the summed scale:

$$\alpha = \frac{k}{k-1}\left[1 - \frac{\sum_j \text{Var}(x_j)}{\text{Var}\left(\sum_j x_j\right)}\right]$$

Dillon-Goldstein's rho and the code's composite reliability calculation use standardized outer loadings lambda_j. With indicator error variance 1 - lambda_j^2:

$$\text{CR} = \frac{\left(\sum_j \lambda_j\right)^2}{\left(\sum_j \lambda_j\right)^2 + \sum_j \left(1 - {\lambda_j}^2\right)}$$

Average variance extracted is the mean squared loading:

$$\text{AVE} = \frac{\sum_j {\lambda_j}^2}{k}$$

CR measures the proportion of composite-score variance attributed to the common construct, while AVE measures the average indicator variance captured by the construct. A single-indicator Stage 2 construct has loading 1 by construction in this implementation, so its apparent perfect reliability should not be interpreted like multi-item reliability.

5.4 PLS-SEM score estimation and path coefficients

PLS-SEM estimates latent scores as weighted combinations of standardized indicators. For construct c with indicator matrix X_c and outer-weight vector w_c:

$$\eta_c = X_c w_c$$

Under reflective Mode A, outer weights are iteratively updated from the association between each indicator and the construct's inner proxy. With path weighting, the proxy combines scores of constructs connected to c. The algorithm alternates outer-score estimation, inner approximation, weight updating, and normalization until the weights stabilize.

After convergence, each endogenous construct is regressed on its predecessor construct scores. For endogenous score eta_y and predictor-score matrix Eta_X, the ordinary least-squares path vector is:

$$\hat{\beta} = \left({H_X}^T H_X\right)^{-1} {H_X}^T \eta_y$$

The Stage 2 higher-order composites are weighted combinations of Stage 1 scores:

$$\text{Relationship Quality} = w_T \text{Trust} + w_L \text{Loyalty}$$

$$\text{Behavioral Response} = w_I \text{Intention} + w_B \text{Behavior}$$

The active serial structural model can be written as:

$$\text{Parasocial} = a\text{Expertise} + \varepsilon_P$$

$$\text{Relationship Quality} = b\text{Parasocial} + \varepsilon_{RQ}$$

$$\text{Behavioral Response} = c\text{Relationship Quality} + \varepsilon_{BR}$$

5.5 Serial mediation and bootstrapping

With no active direct shortcuts in Stage 2, the serial indirect effect from Expertise to Behavioral Response is the product of the three path coefficients:

$$\text{indirect} = abc = 0.743731 \times 0.878555 \times 0.750404 = 0.490320$$

Bootstrapping draws B = 5,000 resamples with replacement, re-estimates the model in each sample, and forms an empirical sampling distribution for each statistic theta. If theta_b* is the result from bootstrap sample b:

$$\text{SE}_{boot} = \sqrt{\frac{\sum_b (\theta^*{}_b - \bar{\theta}^*)^2}{B-1}}$$

$$t_{boot} = \frac{\hat{\theta}}{\text{SE}_{boot}}$$

The reported percentile confidence interval uses the 2.5th and 97.5th percentiles of the 5,000 bootstrap estimates. A percentile interval excluding zero indicates that the corresponding effect is statistically distinguishable from zero at the two-sided 5% level, subject to the bootstrap assumptions.

5.6 Structural and measurement diagnostics

For an endogenous construct y, explained variance is:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

With p predictors, adjusted R-squared penalizes model size:

$$R_{adj}{}^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}$$

Stone-Geisser-style predictive relevance compares prediction error with a naive benchmark:

$$Q^2 = 1 - \frac{\text{PRESS}}{\text{SSO}}$$

Here PRESS is the sum of squared prediction errors and SSO is the corresponding sum of squared deviations from the benchmark. The f-squared effect size measures how much an included predictor contributes to R-squared:

$$f^2 = \frac{R_{included}{}^2 - R_{excluded}{}^2}{1 - R_{included}{}^2}$$

Variance inflation for predictor j uses the R-squared obtained by regressing that predictor on the remaining predictors:

$$\text{VIF}_j = \frac{1}{1 - R_j{}^2}$$

SRMR is the root mean square difference between observed and model-implied standardized correlations. If r_hl and r_hat_hl are the two correlation entries over the q unique entries considered:

$$\text{SRMR} = \sqrt{\frac{\sum_{h \le l} (r_{hl} - \hat{r}_{hl})^2}{q}}$$

HTMT compares the average absolute cross-construct indicator correlations with the geometric mean of the average within-construct correlations:

$$\text{HTMT}_{ab} = \frac{\text{mean}\left(|r_{ij}|, i \in a, j \in b\right)}{\sqrt{\text{mean}(|r_{ii'}|, ii' \in a)\,\text{mean}\left(|r_{jj'}|, jj' \in b\right)}}$$

5.7 Standardization, K-means, and cluster selection

Before clustering, each Stage 2 score is standardized:

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}$$

K-means assigns each respondent to one of K clusters and minimizes within-cluster squared Euclidean distance. If c(i) is respondent i's cluster and mu_k is centroid k:

$$\mathrm{WCSS}(K) = \sum_i \left\| z_i - \mu_{\mathrm{c}(i)} \right\|^2$$

The elbow plot examines diminishing reductions in WCSS as K increases. The silhouette score for respondent i uses a(i), the mean distance to its own cluster, and b(i), the smallest mean distance to another cluster:

$$\mathrm{s}(i) = \frac{\mathrm{b}(i) - \mathrm{a}(i)}{\max\left(\mathrm{a}(i), \mathrm{b}(i)\right)}$$

The overall silhouette is the mean of s(i). Higher values indicate greater separation. The Calinski-Harabasz index compares between-cluster and within-cluster dispersion:

$$\mathrm{CH}(K) = \frac{\frac{B_K}{K-1}}{\frac{W_K}{n-K}}$$

Higher CH is preferred. The Davies-Bouldin index compares within-cluster scatter S_k with centroid separation M_kl:

$$\mathrm{DB}(K) = \frac{1}{K} \sum_k \max_{l \neq k} \frac{S_k + S_l}{M_{kl}}$$

Lower DB is preferred. The attached run selects K = 2 because it simultaneously has the highest silhouette and CH values and the lowest DB value among the tested solutions. Cluster-profile means are calculated as n_k^(-1) sum_{i in k} z_i, and the Low/High Engagement labels are assigned after inspecting the signs and magnitudes of those profiles.

5.8 Decision-tree mathematics

The classifier recursively partitions the predictor space. For a node t containing class proportions p_kt, the default scikit-learn classification impurity is the Gini index:

$$\mathrm{Gini}(t) = 1 - \sum_k {p_{kt}}^2$$

For a candidate split into left and right child nodes, the weighted impurity reduction is:

$$\Delta G = \mathrm{Gini}(t) - \frac{n_L}{n_t}\mathrm{Gini}(L) - \frac{n_R}{n_t}\mathrm{Gini}(R)$$

At each node, the algorithm selects the split with the largest impurity reduction, subject to max_depth = 4 and min_samples_leaf = 10. A feature's reported importance is its normalized sum of weighted impurity decreases across every node that uses that feature:

$$\text{Importance}(j) = \sum_{t \text{splits on } j} \frac{n_t}{n} \Delta G_t, \text{normalized so} \sum_j \text{Importance}(j) = 1$$

This quantity describes how strongly a variable contributed to the fitted tree. It is not a causal effect, and it can be biased when coded categories are treated as ordered numbers or when predictors differ in the number of available split points.

6. Generated artifacts

Artifact	Created by	Purpose
output.csv	Source_Code1	Cleaned and imputed 249-row dataset consumed by Source_Code2
correlation_heatmap.png	Source_Code1	300 dpi lower-triangular Pearson-correlation heatmap
figure.eps	Source_Code1	Vector version of the labeled correlation heatmap
decision_tree_engagement_font20.png	Source_Code2	High-resolution engagement-cluster decision tree
decision_tree_engagement_font20.pdf	Source_Code2	PDF version of the decision tree
decision_tree_font35.png	Source_Code2	Alternative tree image with larger text
decision_tree_engagement_font35.pdf	Source_Code2	PDF version of the larger-text tree

Appendix A. Quick object reference

Object	Created in	Meaning
df	Both notebooks	Main respondent DataFrame
dq	Source_Code1	Q1-Q34 subset used for response-quality checks
constructs / cons	Source_Code1	Indicator map and construct-name list
pls	Both notebooks	First-order openpls Plspm model
scores	Both notebooks	First-order latent-variable scores
stage2_df	Source_Code2	Copy of Stage 1 scores used as Stage 2 data
pls_stage2	Source_Code2	Higher-order serial PLS-SEM model
Fscores	Source_Code2	Stage 2 scores later used for clustering
X_scaled	Source_Code2	Standardized four-construct clustering matrix
df_model	Source_Code2	Cleaned data with cluster number and label
tree	Source_Code2	Fitted demographic/usage decision-tree classifier