**House of Travel — Total Base Analysis (Part 1)  
Handover Documentation**

***Prepared by David Airehrour | Together***

# 1. Purpose & Scope

This document is the handover guide for Part 1 of the House of Travel (HoT) customer analytics project: **the Total Base analysis**.   
It explains the end‑to‑end logic implemented in the Python script, key business rules, deduplication and profile hygiene (SPiD logic), feature engineering for downstream segmentation, and the outputs delivered to House of Travel.

# 2. Repository & Key Assets

* GitHub: https://github.com/Together-NZ/HoT-Totalbase-Analysis
* Primary script: hot-totabase-analysis.ipynb
* Reference notes: SPiD Logic (redundant/duplicate profile rules) and Dolphin BMM – Merging Duplicate Profiles.

# 3. Data Inputs

**3.1 Customer exports (Salesforce Marketing Cloud extracts via GCS)**

* Bucket: hot-together-dropbox (Google Cloud Storage).
* Files: HOT\_SFMC\_Customer\_Incremental\_2025-02-19.csv and HOT\_SFMC\_Customer\_Incremental\_2024-12-04.csv.
* Delimiter: pipe (|), Encoding: latin1, with on\_bad\_lines='skip' to safely ignore malformed rows.
* Filter: only profiles with ProfileState == 'Live' and drop rows with missing ClientID or UnsubscribeURL; cast ClientID to int.

**3.2 Transactional exports (Folder/booking history)**

* Bucket: hot-together-dropbox (Google Cloud Storage).
* Files: HOT\_SFMC\_Transactional\_Incremental\_2024-12-04.csv and HOT\_SFMC\_Transactional\_Incremental\_Folder Data\_2025-02-19.csv.
* Delimiter: pipe (|), Encoding: latin1; cast ClientID to int.
* Filter to confirmed revenue: FolderStatus == 'Invoiced', FolderValue > 0, and FolderConfirmationDate between 2022‑01‑01 and 2024‑12‑31.

# 4. Profile Hygiene & Deduplication (SPiD Logic)

**4.1 Redundant profile deletion — business rules**

* Profiles (head and linked) created before 01‑Jan‑2015, with no email or email‑info only, and
* NO confirmed folders in the last 5 years, and
* NO quote/void folders in last 3 years, and
* NO special‑interest codes are marked Deleted.

**4.2 Duplicate identification (Dolphin matching)**

* Match on LastName & FirstName (or first three initials), plus ONE of: email, DOB, phone (last 7 digits, numeric‑only), address line 1, passport number & nationality.
* **We have used email as this is the most accurate and has over 90% non-null values.**

**4.3 In‑script dedup strategy**

* Combine latest and historical customer extracts; drop duplicates by ClientID (keep first).
* Create a temporary name‑email group\_key (FirstName|LastName|EmailAddress) to surface RelatedClientID sets (candidates for SPiD merge review).
* Fill missing EmailAddress from EmailInfoAddress when available.
* Calculate client\_age from Birthdate; then drop Birthdate to reduce PII propagation.
* Clean FrequentFlyerDetails into a concise ff\_code (vendor/airline code list).

# 5. Feature Engineering

**5.1 Communication & identifiers**

* Binary flags: EmailFlag, MMEmailFlag, HOOTEmailFlag, SkiEmailFlag, CruiseEmailFlag, LinkedToCompany.
* Channel/app fields: MMLogin, AppDownloadHOT (retain as indicators for engaged membership).
* Frequent flyer: ff\_code (parsed/cleaned from FrequentFlyerDetails).

**5.2 Travel product repertoire**

* Has\_\* flags derived per folder line items: Has\_Air, Has\_Hotel, Has\_Tour, Has\_Cruise, Has\_Insurance.
* CuratedSegments: consolidated set of product categories for repertoire and indexing analytics.

**5.3 Demographic & party structure**

* client\_age from Birthdate; age\_band bucketing for reporting.
* PaxNumber and AgeOfOtherTravellers used to derive family structures (adult/teen/child).
* Travel group categorisation helpers: Younger Family, Older Family, Sinks & Dinks, Empty Nester, Solo‑based segments.

**6.4 Geography & destination tags**

* Map Destinations to DestinationRegion/continent; flag specialty destinations (e.g., Antarctica, Galápagos).
* ArrivalAirport flags for Anchorage (AK), Rarotonga, Bali, etc., for proxying specific travel interests.

**6.5 Business vs Leisure signal**

* SME\_Corporate flag from LinkedToCompany==1 (corporate/business travel).
* Leisure‑focused segments exclude SME\_Corporate records to avoid contaminating audience builds.

# 7. Segmentation Logic (Part 1: Total Base)

**7.1 Value Segments (customer‑level)**

* Aggregate per ClientID: TotalFolderValue (sum of FolderValue) and total\_transactions (count of FolderID).
* Thresholds on TotalFolderValue: HVC (top 20%), MVC (next 30%), LVC (bottom 50%).
* Map ValueSegment back to merged\_df for downstream analysis and indexing.

**7.2 Behavioural & life‑stage segments (folder‑level, then rolled to client)**

* Air Only: Air present, no Hotel/Tour/Cruise (Insurance optional), exclude cruise entirely.
* Luxury Seeker: luxury vendor signals (hotel/cruise/tour lists) OR high FolderValue per pax rules; exclude SME\_Corporate.
* Yolo Solo: PaxNumber==1, client\_age ≥ 41, minimal components (no Cruise), exclude Air Only and SME\_Corporate.
* Young Explorer: client\_age 21–40 with minimal components OR (missing DOB + youth tour brands such as Contiki/Top Deck/Sail Croatia); exclude Air Only/SME\_Corporate.
* Younger Family: Pax ≥ 2, at least one adult (>21) and one child (<12), and NO teens (12–21); exclude SME\_Corporate.
* Older Family: Pax ≥ 2, at least one adult (>21) and at least one teen (12–21); exclude SME\_Corporate.
* Sinks & Dinks: Pax==2 and age 25–54; exclude SME\_Corporate.
* Empty Nester: Pax==2 and age ≥ 55; exclude SME\_Corporate.

**7.3 Output artefacts**

* Per‑segment indexing workbooks saved to: d:/downloads/hot-segments-total-base/<Segment>\_indexing.xlsx (adjust path as required).
* Outlet/region summaries and SME\_Corporate extracts also written to d:/downloads/… as CSV/XLSX in the script.
* Intersection matrix (optional) for segment overlaps can be exported (commented in script).

# 8. How to Run

1. Set up Google credentials (Application Default Credentials) with access to the GCS bucket: hot-together-dropbox.
2. Update file names if newer extracts are available (customer and transactional CSVs).
3. Install Python dependencies (pandas, numpy, matplotlib, google‑cloud‑storage).
4. Execute hot-totabase-analysis.py in your preferred environment (VS Code/Jupyter).
5. Confirm row counts after dedup; inspect RelatedClientID to triage merges (SPiD‑guided).
6. Validate confirmed\_txn\_df filters (FolderStatus, value, and date window).
7. Review segment counts and index sheets in the output directory; align any vendor lists/thresholds with Marketing.

# 9. Key Nuances & Gotchas

* Email hygiene: fill EmailAddress from EmailInfoAddress when missing to improve de‑dup grouping.
* Age derivation: Birthdate → client\_age then drop Birthdate to minimise PII exposure downstream.
* Frequent flyer normalisation: extract a consistent ff\_code list and strip noisy tokens ('\*', '00', '?', single letters).
* Corporate leakage: exclude LinkedToCompany==1 from leisure segments to avoid mis‑targeting.
* Cruise logic: many non‑cruise segments explicitly exclude clients with Cruise components.
* Vendor lists: Luxury and youth‑brand lists are curated; keep them in a constants block for easy refresh.
* Time window: The analysis window (2022–2024) is hard‑coded for stability; revisit quarterly for recency.
* Output paths: update local 'd:/downloads/…' paths for your environment (e.g., to a shared drive).

# 10. Handover Checklist

* ✅ Access to GCS bucket and latest CSVs verified.
* ✅ GitHub repo cloned and script paths updated.
* ✅ SPiD dedup review completed (profiles merged/deleted as per rules).
* ✅ Segment counts sanity‑checked; output workbooks delivered to Sales/CRM.
* ✅ Documented any deviations (threshold tweaks, vendor list updates, or outlet exclusions).

# Appendix A — Representative Code Snippets

**A1. Customer ingest**

# Read customer extract (GCS) with safe defaults  
customer\_df = pd.read\_csv(temp\_file.name, sep='|', encoding='latin1').query("ProfileState=='Live'")  
customer\_df = customer\_df.dropna(subset=['ClientID','UnsubscribeURL'])  
customer\_df['ClientID'] = customer\_df['ClientID'].astype(int)

**A2. Customer deduplication**

# Combine old/new customer extracts & remove duplicate ClientIDs  
customer\_df\_deduped = customer\_df.drop\_duplicates(subset=['ClientID'], keep='first')  
customer\_df\_new = pd.concat([customer\_df\_old, customer\_df\_deduped], axis=0)\  
 .drop\_duplicates(subset=['ClientID'], keep='first').reset\_index(drop=True)

**A3. SPiD pre‑merge discovery**

# RelatedClientID discovery (name+email group)  
def make\_group\_key(row):  
 fn = str(row['FirstName']) if pd.notnull(row['FirstName']) else ""  
 ln = str(row['LastName']) if pd.notnull(row['LastName']) else ""  
 em = str(row['EmailAddress']) if pd.notnull(row['EmailAddress']) else ""  
 return fn + "|" + ln + "|" + em

**A4. Transaction filter**

# Confirmed transactions window  
confirmed\_txn\_df = transaction\_df[  
 (transaction\_df['FolderStatus'] == 'Invoiced') &  
 (transaction\_df['FolderValue'] > 0) &  
 (pd.to\_datetime(transaction\_df['FolderConfirmationDate']).between('2022-01-01','2024-12-31'))  
]

**A5. HVC/MVC/LVC thresholds**

# Value segmentation  
customer\_summary = (hvc\_profile.groupby('ClientID')  
 .agg({'FolderValue':'sum','FolderID':'count'})  
 .rename(columns={'FolderValue':'TotalFolderValue','FolderID':'total\_transactions'})  
 .reset\_index())  
hvc\_threshold = customer\_summary['TotalFolderValue'].quantile(0.8)  
mvc\_threshold = customer\_summary['TotalFolderValue'].quantile(0.5)

# Appendix B — Future (Part 2: Leisure Base)

* Extend segments with leisure‑specific nuances (e.g., destination themes, seasonality, brand affinity).
* Add RFM quartiles per product line; compute cross‑sell propensity features by repertoire.
* Publish activation‑ready audiences with privacy‑by‑design controls (hashing, suppression, unsubscribe logic).