Excel Guide: SKU Gaps & Upsell Readiness Score

This comprehensive guide provides a practical, step-by-step approach to creating an Excel sheet for tracking SKU gaps and calculating an upsell readiness score. It cuts through the noise, offering only essential setup instructions, key assumptions, and precise formulas with clear explanations for each requirement. No fluff—just what you need to set up, the assumptions, and the exact formulas with explanations of each requirement.

1. Sheet & Table Structure

1. **Purchases table**
    - Create a sheet (e.g., named “Purchases”) with at least these columns in row 1:
        - `CustomerID`
        - `SKU`
        - `PurchaseDate`
        - `Amount` (or Revenue)
    - Convert that range to an Excel Table: select the data, Insert → Table. Name it `Purchases` (use the Table Name box in Table Design).
    - **Why**: structured references (e.g., `Purchases[CustomerID]`) auto-expand when new rows are added and make formulas clearer.

2. **Master SKU list**
    - On another sheet (e.g., “SKUList”), have a single-column table:
        - Column header: `SKU`
        - Rows: every SKU you track.
    - Convert to Table, name it `SKUList`.
    - **Why**: you need a complete list of SKUs to check which ones a customer has or hasn’t bought.

3. **Customer list / Summary sheet**
    - On a “Summary” sheet, list each `CustomerID` in column A, starting A2 downward.
    - You can convert this to a Table named `Customers`, but it’s optional. The key is that in row 2 you refer to CustomerID cell as `$A2` when writing formulas.

4. **(Optional) Threshold/config table**
    - If you want flexible buckets (for recency, frequency, monetary thresholds), create a small table “Thresholds” with named ranges, e.g.:
        - RecencyDays / Score mapping
        - FrequencyCount / Score mapping
        - MonetaryAmount / Score mapping

- Then use LOOKUP or INDEX/MATCH instead of hard-coded IFS. This saves future tuning. But you can start with hard-coded buckets.

2. Identifying SKU Coverage Gaps

Goal

For each customer, flag every SKU as “Owned” or “Gap” (never purchased). Then compute penetration rate or count of gaps.

Setup on Summary sheet

- Column A: `CustomerID` (e.g., A2 = the first CustomerID to evaluate).
- Columns B onward: list SKUs (one per row), or better: put SKUs in column B on a separate “CustomerSKU” area. But more common is:
  - On the “Summary” sheet, in B1 put header `SKU`, in C1 put header `GapFlag`.
  - In B2 downward, paste all SKUs from SKUList (you can reference SKUList table or use a lookup). For simplicity: in B2: `=INDEX(SKUList[SKU], ROW()-1)` if you know SKUList starts at SKUList[SKU][#All], but you can also copy-paste the SKU list.
  - In A2 (CustomerID), you keep that fixed for the block of SKUs. If you want one sheet per customer, you could have CustomerID in a cell and SKUs in a column below.

Formula to flag gap vs owned

In C2 (header "GapFlag"), use:

```
Unset
=IF(
  COUNTIFS(
    Purchases[CustomerID], $A$2,
    Purchases[SKU], B2
  ) = 0,
  "Gap",
  "Owned"
)
```

- **Requirements**:
  - `Purchases` table exists with columns exactly named `CustomerID` and `SKU`.
  - `$A$2` holds the CustomerID you evaluate.
  - `B2` holds one SKU from the master list.
- Copy/fill down C2 for all SKUs in B2:B(n).
- **Interpretation**:

- If count of purchase rows where CustomerID matches AND SKU matches is zero → “Gap”; otherwise “Owned”.

Summarize counts / rate

Some cells on the Summary sheet (outside the SKU list block):

- **Total SKUs**: if SKUs in B2:B101, then `=COUNTA(B2:B101)`.
- **Owned count**: `=COUNTIF(C2:C101, "Owned")`.
- **Gap count**: `=COUNTIF(C2:C101, "Gap")`.
- **Penetration rate**:

```
Unset
=COUNTIF(C2:C101, "Owned") / COUNTA(B2:B101)
```

- Format as percentage.

Alternative without helper column

If you don’t want a “GapFlag” column, you can directly compute gap count in one formula:

```
Unset
=SUMPRODUCT(
  --(
    COUNTIFS(
      Purchases[CustomerID], $A2,
      Purchases[SKU], SKUList[SKU]
    ) = 0
  )
)
```

- **Requirements**:
  - `SKUList` table with column `SKU`.
  - `Customers` or just CustomerID in `$A2`.
- This returns the number of SKUs never purchased by that customer. Then

```
Unset
Penetration = (COUNTA(SKUList[SKU]) - this) /
COUNTA(SKUList[SKU])
```

---

3. Building Upsell Readiness Score

We'll derive multiple metrics per customer, normalize each into a small numeric score (e.g., 1–5), then weight-combine.

3.1 Base metrics per customer

On "Summary" sheet, assume:

- Column A: `CustomerID` (A2).
- Column B: `LastPurchaseDate`.
- Column C: `TotalPurchasesCount`.
- Column D: `TotalRevenue`.
- Column E: `DistinctSKUCount`.
- Column F: `DaysSinceLastPurchase`.
- (Additional columns for trend if desired.)

3.1.1 LastPurchaseDate

In B2:

```
Unset
=IFERROR(
  MAXIFS(Purchases[PurchaseDate], Purchases[CustomerID], $A2),
  ""  /* or NA() if you prefer */
)
```

- **Requirements**:
  - `PurchaseDate` column in `Purchases` is proper Excel dates.
  - If no purchases, MAXIFS returns error; we wrap IFERROR.

3.1.2 TotalPurchasesCount

In C2:

```
Unset
=COUNTIFS(Purchases[CustomerID], $A2)
```

- Counts all purchase rows for that customer.

3.1.3 TotalRevenue

In D2:

```
Unset
=SUMIFS(Purchases[Amount], Purchases[CustomerID], $A2)
```

- Sum of Amount for that customer.

3.1.4 DistinctSKUCount

Option A: Use helper pivot or advanced formulas. In formula:

```
Unset
=SUMPRODUCT(
  --(
    COUNTIFS(
      Purchases[CustomerID], $A2,
      Purchases[SKU], SKUList[SKU]
    ) > 0
  )
)
```

- **Requirements**:
  - `SKUList` table exists listing all SKUs.
  - This counts how many SKUs from master list were purchased at least once by this customer.
- If your SKUList is huge and Purchases is large, this can be slow. In that case, consider pivot or helper columns in Purchases to mark unique combos, then COUNTIFS on that helper.

3.1.5 DaysSinceLastPurchase

In F2:

```
Unset
=IF(B2="", "", TODAY() - B2)
```

- **Requirements**:
  - B2 must be a date or blank.

3.1.6 Trend signals (optional)

E.g., spending last 3 months vs prior 3 months:

- Define dates:
  - `StartRecent` = TODAY() - 90
  - `StartPrior` = TODAY() - 180
- Recent spend:

```
Unset
=SUMIFS(
  Purchases[Amount],
```

```
  Purchases[CustomerID], $A2,
  Purchases[PurchaseDate], ">" & TODAY()-90
)
```

- Prior spend:

```
Unset
=SUMIFS(
  Purchases[Amount],
  Purchases[CustomerID], $A2,
  Purchases[PurchaseDate], ">" & TODAY()-180,
  Purchases[PurchaseDate], "<=" & TODAY()-90
)
```

- Then trend ratio or difference:

```
Unset
=IF(Prior=0, "", Recent/Prior)
```

- **Requirements**: date arithmetic is correct; wrap IFERROR as needed.

3.2 Normalizing metrics into scores (1–5)

Decide bucket thresholds based on your data distribution. You can hard-code in IFS or use lookup tables. Example with IFS:

3.2.1 RecencyScore (column G)

In G2:

```
Unset
=IFS(
  F2 = "", 1,                  /* no purchases = lowest score */
  F2 <= 30, 5,
  F2 <= 60, 4,
  F2 <= 90, 3,
  F2 <= 180, 2,
  TRUE, 1
)
```

- **Requirements**: F2 (DaysSinceLastPurchase) numeric or blank.

3.2.2 FrequencyScore (column H)

In H2, based on C2:

```
Unset
=IFS(
  C2 >= 20, 5,
  C2 >= 10, 4,
  C2 >= 5, 3,
  C2 >= 2, 2,
  TRUE, 1
)
```

- **Requirements**: adjust thresholds (20,10,5,2) to your business reality.

3.2.3 MonetaryScore (column I)

In I2:

```
Unset
=IFS(
  D2 >= 10000, 5,
  D2 >= 5000, 4,
  D2 >= 2000, 3,
  D2 >= 500, 2,
  TRUE, 1
)
```

- **Requirements**: tune amounts to your pricing/industry.

3.2.4 PenetrationGapScore (column J)

We want a higher score if penetration rate is low (many gaps) but maybe high spend. Example:

- Compute penetration rate in a hidden column or inline:
  - PenetrationRate = `DistinctSKUCount / COUNTA(SKUList[SKU])`.
- In J2:

```
Unset
=LET(
  totalSKUs, COUNTA(SKUList[SKU]),
```

```
  distinct, E2,  /* if you stored DistinctSKUCount in E2;
otherwise recalc */
  penetration, IF(totalSKUs=0, 1, distinct/totalSKUs),
  IFS(
    penetration < 0.2, 5,
    penetration < 0.5, 3,
    TRUE, 1
  )
)
```

- **Requirements**:
  - E2 holds DistinctSKUCount (or inline recalc).
  - SKUList table exists.
- Interpretation: if they've bought <20% of SKUs → high upsell potential = 5; between 20–50% → medium = 3; >50% → low = 1. Tweak as needed.

3.3 Combine into a final Upsell Readiness Score

On column K (UpsellScoreNumeric):

- Choose weights. Example: Recency 30%, Frequency 20%, Monetary 25%, PenetrationGap 25%.
- In K2:

```
Unset
=G2*0.3 + H2*0.2 + I2*0.25 + J2*0.25
```

- **Requirements**: G2, H2, I2, J2 are numeric scores 1–5.
- This yields a number between 1 and 5. You can leave as-is or bucket again.

Bucket into descriptive label (column L):

In L2:

```
Unset
=IFS(
  K2 >= 4.5, "Very High",
  K2 >= 3.5, "High",
  K2 >= 2.5, "Medium",
  TRUE, "Low"
)
```

- **Requirements**: adjust cutoffs if you want a different distribution.

4. Step-by-Step: Building your own sheet

1. **Create Purchases table**
    - Gather raw data exports from your system (CSV export of orders/invoices). Ensure columns: CustomerID, SKU, PurchaseDate, Amount.
    - Paste into “Purchases” sheet, convert to Table named `Purchases`. Check that PurchaseDate is date type.

2. **Create SKUList table**
    - On “SKUList” sheet, paste the full list of SKUs (unique). Convert to Table named `SKUList`.

3. **Create Summary sheet**
    - Column A: paste unique CustomerIDs (or link from a Customers table). In A1 header “CustomerID”.
    - Next columns: set up headings as described:
        - B1: `LastPurchaseDate`
        - C1: `TotalPurchasesCount`
        - D1: `TotalRevenue`
        - E1: `DistinctSKUCount`
        - F1: `DaysSinceLastPurchase`
        - G1: `RecencyScore`
        - H1: `FrequencyScore`
        - I1: `MonetaryScore`
        - J1: `PenetrationGapScore`
        - K1: `UpsellScoreNumeric`
        - L1: `UpsellCategory`
    - In row 2, enter formulas above. Then fill down for all customers.

4. **(Optional) SKU Gap matrix block**
    - If you also want a per-customer per-SKU flag table, create a separate sheet “GapMatrix”:
        - Column A: CustomerID (repeat per block or use pivot).
        - Column B: SKU.
        - Column C: GapFlag formula referencing Purchases. Or use a pivot: create pivot with CustomerID rows, SKU columns, Values = COUNT of PurchaseDate; blanks indicate gaps.
    - But often the summary metrics suffice; you don’t need to list every SKU in the sheet unless you need a detailed report.

5. **Tune threshold buckets**

- Review distributions: e.g., add a quick pivot or sort on TotalPurchasesCount to see typical ranges. Adjust IFS thresholds accordingly.
- If you created a “Thresholds” table, update that instead of editing formulas.

6. **Test on a few customers**
   - Pick 3–5 known accounts: verify LastPurchaseDate is correct, TotalRevenue matches manual check, DistinctSKUCount plausible, and final score aligns with your intuition.

7. **Performance considerations**
   - If Purchases table has thousands of rows and SKUList is large, COUNTIFS inside SUMPRODUCT can get slow. In that case:
     - Use a PivotTable: rows = CustomerID, columns = SKU, values = Count of orders. Then in the pivot you can see where counts are zero. Or:
     - Add a helper column in Purchases combining CustomerID & SKU, then use UNIQUE + COUNTIF to get distinct SKUs per customer more efficiently.
   - Consider Power Query: import Purchases, group by CustomerID to compute metrics in one go, then load results to Summary sheet. Power Query can handle large data more smoothly.

8. **Automate data refresh**
   - If your Purchases table is an export, consider linking via Power Query to your source or a regularly updated CSV so your Summary recalculates when data updates.

9. **Documentation on your sheet**
   - In a “README” sheet, note:
     - Table names: Purchases, SKUList.
     - Assumptions: PurchaseDate is date; Amount is numeric; CustomerID and SKU exact matches.
     - How to add new data: append rows to Purchases table.
     - How to adjust thresholds: edit formulas or Thresholds table.
     - How to interpret UpsellScore: numeric 1–5 scale combined into categories via bucket.

5. Example formulas recap (with structured references)

On Summary row 2, where A2 = CustomerID:

```
Unset
B2: LastPurchaseDate
```

```
=IFERROR(MAXIFS(Purchases[PurchaseDate], Purchases[CustomerID],
$A2), "")

C2: TotalPurchasesCount
=COUNTIFS(Purchases[CustomerID], $A2)

D2: TotalRevenue
=SUMIFS(Purchases[Amount], Purchases[CustomerID], $A2)

E2: DistinctSKUCount
=SUMPRODUCT(--(COUNTIFS(Purchases[CustomerID], $A2,
Purchases[SKU], SKUList[SKU])>0))

F2: DaysSinceLastPurchase
=IF(B2="", "", TODAY()-B2)

G2: RecencyScore
=IFS(
  F2="", 1,
  F2<=30, 5,
  F2<=60, 4,
  F2<=90, 3,
  F2<=180, 2,
  TRUE, 1
)

H2: FrequencyScore
=IFS(
  C2>=20,5,
  C2>=10,4,
  C2>=5,3,
  C2>=2,2,
  TRUE,1
)

I2: MonetaryScore
```

```
=IFS(
  D2>=10000,5,
  D2>=5000,4,
  D2>=2000,3,
  D2>=500,2,
  TRUE,1
)

J2: PenetrationGapScore
=LET(
  totalSKUs, COUNTA(SKUList[SKU]),
  distinct, E2,
  penetration, IF(totalSKUs=0,1, distinct/totalSKUs),
  IFS(
    penetration<0.2,5,
    penetration<0.5,3,
    TRUE,1
  )
)

K2: UpsellScoreNumeric
=G2*0.3 + H2*0.2 + I2*0.25 + J2*0.25

L2: UpsellCategory
=IFS(
  K2>=4.5, "Very High",
  K2>=3.5, "High",
  K2>=2.5, "Medium",
  TRUE, "Low"
)
```

- Adjust the numeric thresholds (e.g., days, counts, amounts, penetration cutoffs) to match your data's reality.
- If you prefer structured flexibility, replace IFS(...) with LOOKUP against a small threshold table.

Final advice

- **Start small**: build this sheet for a subset of customers first. Confirm formulas, performance, and interpretation.
- **Keep formulas readable**: separate intermediate metrics (LastPurchaseDate, counts) into their own columns rather than cramming everything in one cell.
- **Document assumptions**: e.g., what counts as a “purchase” (any invoice? only paid invoices?), how you treat returns/refunds.
- **Plan for scale**: once the sheet grows slow or you need team access, move to Power Query or a BI/dashboard tool that computes these metrics centrally and surfaces results in a dashboard rather than manual Excel maintenance.
- **Review regularly**: bucket thresholds may need adjustment as your business evolves; keep them in a config area or document where you can update without hunting formula text.

That’s it. With these steps and formulas, you can build your own sheet, understand exactly what each formula requires (table names, column names, data types), and maintain or tune it over time.