1- EKPI Development (D Component)

1-1- D1: Brand SOV in Google (SOVG & SOVGR)

Brand Share of Voice in Google (SOVG) measures the relative visibility of a brand compared to its competitors based on normalized Google Search index counts. Growth rates (SOVGR) capture temporal changes in competitive visibility.
The annual value of SOVG and Growth rates (SOVGR) are implemented with the following DAX formula:

```
CombinedValues = //SOVG
SUMX(
    FILTER(
        SOVG,
                SOVG[Keyword] = EARLIER(SOVG[Keyword]) &&
        SOVG[Year] = EARLIER(SOVG[Year])
    ),
    SOVG[search_result]
)
```

```
CAGR_inYear = //SOVGR
VAR CurrentKeyword = 'SOVG'[Keyword]
VAR CurrentYear = YEAR('SOVG'[Date])
VAR StartDate =
    CALCULATE(
        MIN('SOVG'[Date]),
        FILTER(
            'SOVG',
            'SOVG'[Keyword] = CurrentKeyword &&
            YEAR('SOVG'[Date]) = CurrentYear))
VAR EndDate =
    CALCULATE(
        MAX('SOVG'[Date]),
        FILTER(
            'SOVG',
            'SOVG'[Keyword] = CurrentKeyword &&
            YEAR('SOVG'[Date]) = CurrentYear))

VAR Value_Start =
    CALCULATE(
        AVERAGE('SOVG'[NormalResult]),
        FILTER('SOVG', 'SOVG'[Keyword] = CurrentKeyword && 'SOVG'[Date] = StartDate))
VAR Value_End =
    CALCULATE(
        AVERAGE('SOVG'[NormalResult]),
        FILTER('SOVG', 'SOVG'[Keyword] = CurrentKeyword && 'SOVG'[Date] = EndDate))
RETURN
IF(
    NOT ISBLANK(Value_Start) && NOT ISBLANK(Value_End) && Value_Start > 0,
    ((Value_End / Value_Start) - 1) * 100,
    BLANK()
)
```

1-2- D2: Brand Awareness in Google (BAG & BAGR)

Brand Awareness in Google (BAG) is derived from normalized Google Trends timeline data and reflects aggregate public interest in a brand. Corresponding growth rates (BAGR) capture changes in awareness intensity over time. The Growth rates are implemented with the following DAX formula:

```
CombinedIrancell = //BAG (For each brand separately)
SUMX(
    FILTER(
        Timeline,
        Timeline[Year] = EARLIER(Timeline[Year])
    ),
    Timeline[AwIrancell]
)
```

```
CAGR_AwIrancell = //BAGR (For each brand separately)
VAR CurrentYear = YEAR('Timeline'[Date])
VAR StartDate =
    CALCULATE(
        MIN('Timeline'[Date]),
        FILTER(
            'Timeline',
            YEAR('Timeline'[Date]) = CurrentYear))
VAR EndDate =
    CALCULATE(
        MAX('Timeline'[Date]),
        FILTER(
            'Timeline',
            YEAR('Timeline'[Date]) = CurrentYear))
VAR Value_Start =
    CALCULATE(
        AVERAGE('Timeline'[AwIrancell]),
        FILTER('Timeline',
            'Timeline'[Date] = StartDate))
VAR Value_End =
    CALCULATE(
        AVERAGE('Timeline'[AwIrancell]),
        FILTER('Timeline',
            'Timeline'[Date] = EndDate))
RETURN
IF(
    NOT ISBLANK(Value_Start) &&
    NOT ISBLANK(Value_End) &&
    Value_Start > 0,
    ((Value_End / Value_Start) - 1) * 100,
    BLANK()
)
```

1-3- D3: Brand Sentiment in Google Trends (PS-NS & PSR)

Brand sentiment indicators are computed from the distribution of positive and negative related queries. Net sentiment (PS–NS) reflects the balance between favorable and unfavorable user intent, while the positive sentiment ratio (PSR) captures the proportion of positive sentiment among all sentiment-bearing queries. The PS, NS, and PSR are implemented with the following DAX formula:

```
PSR =
VAR CurrentYear = Sentiment[Year]
VAR CurrentBrand = Sentiment[Keyword]
VAR SumPositive = //PS
    CALCULATE(
        SUM(Sentiment[CombinedValues]),
        ALLEXCEPT(Sentiment, Sentiment[Year], Sentiment[Keyword]),
        Sentiment[Sentiment] = "positive")
VAR SumNegative = //NS
    CALCULATE(
        SUM(Sentiment[CombinedValues]),
        ALLEXCEPT(Sentiment, Sentiment[Year], Sentiment[Keyword]),
        Sentiment[Sentiment] = "negative")
VAR Total = SumPositive + SumNegative
RETURN
IF(
    Total > 0,
    DIVIDE(SumPositive, Total),
    BLANK()
)
```

1-4- Brand Position in Google (BPG) (D4)

The Brand Position in Google (BPG) index aggregates all EKPIs into a composite indicator representing overall digital brand positioning. Aggregation is performed using a weighted linear combination, with criterion weights elicited via the Best–Worst Method. Final weights are reported in the main manuscript.

All EKPIs that did not have a normalized range of 1 to 100 after calculation (such as SOVGR, BAGR, and PSR) were each jointly normalized from 1 to 100 before being aggregated into BPG to ensure comparability between heterogeneous indices. BPG is implemented with coefficients using the following DAX formula:

```
BPG_Irancell =
0.01  *(444*[PSR_Irancell] + 184*([PS_Irancell]-[NS_Irancell])+
0.02  138*[BAGR_Irancell]+110*[Aw_Irancell]+
0.03  79*[SOVG_Irancell]+41*[SOVGR_Irancell])
```

2- EKPI and DAX Calculations

Final EKPI and BPG visualizations were produced using Microsoft Power BI to support interactive exploratory analysis. Core computations within the dashboards were implemented using DAX formulas.
To ensure transparency without introducing dependencies on proprietary software, all DAX formulas are documented in text form, while executable `.pbix` files are not distributed.

3- Reproducibility Strategy and Relation to Figure 4

The reproducibility strategy adopted in this study follows a **hybrid model**, combining:

- Fully reproducible open-source code for core analytical steps, and
- Explicit documentation for expert-driven and platform-dependent procedures.

This design reflects the practical realities of applied behavioral analytics, where certain steps (e.g., expert judgment, proprietary tools, and platform-governed data access) cannot be fully automated without compromising validity or compliance.
Importantly, all transformations affecting the reported results are either implemented in code or explicitly documented, ensuring that the analytical logic and decision paths are fully transparent.

Each component of the process model presented in Figure 4 is explicitly mapped to either:

- a Python script or Jupyter notebook in the public repository, or
- A documented manual procedure (for example, DAX or expert judgment) is described in this appendix.

This mapping ensures end-to-end traceability of the EKPI development pipeline while preserving conceptual clarity in the main manuscript.