Behavioural information about each user that can be used to infer user's interests are provided as table datasets - SearchInfo, AdsInfo, trainsearchstream, phonerequestsstream, visitsstream.

**SearchInfo** - details of searches performed by user includes search query, search parameters, userID, categoryID, LocationID, IP address of user- IPID.

One search per row.

Can be joined to category, location, UserInfo

Level in Location (once you join SearchInfo and Location on SearchLocationID = LocationID) represents geo level where a search (and impression) took place. Values could be 3 (city), 2 (region), or 1 (whole country).

Level in Category (once you join SearchInfo and Category on SearchCategoryID = CategoryID) represents category level where a search (and impression) took place. Values could be 3 (subcategory), 2 (category), or 1 (no specific category or subcategory were selected by visitor).

Join with UserInfo on UserID

*-Simulation of SearchInfo data*

-Bronze layer staging table

-Silver - Joining with tables and enriching the data

**AdsInfo** - Data of all the ads ie title/description, parameters, LocationID, categoryID, price, iscontext. Iscontext indicates whether it is contextual ad. Though this is master data, this has 37 million records.

Can be joined to category, location

Level in Location (once you join AdsInfo and Location using LocationID) represents geo level belongs to a non-contextual ad. It has to be 3 (city). Note that for context ad LocationID is NULL.

Level in Category (once you join AdsInfo and Category using CategoryID) represents category level belongs to any ad. Values could be either 3 (subcategory, for all non-contextual and some contextual ads), or 2 (category, for some contextual ads).

*-Simulation of AdsInfo data*

-Bronze layer staging table

-Silver - Joining with tables and enriching the data

Location - Add Geo Level for non-contextual Ad – 3 (city). Location is null for contextual Ad. Else location id has value

Category – 3 (subcategory, for all non-contextual and some contextual ads) or 2 (category, for some contextual ads).

**Category** - Master table of all categories. Level = 1/2/3

**Location** - Master table of all locations. Level = 1/2/3

**UserInfo** - Master table of all users

**trainsearchstream** - Each record describes one "impression" (an ad that is shown to a particular user based on a search). Fields are as follows:

SearchID - identifier for a visitors's search event.

AdID - identifier of the ad (see also ad description in AdsInfo.tsv).

Position - position of the ad in search result page (1 - is first ad on a page starting from the top). Only ads on position 1, 2, 6, 7, and 8 are logged.

ObjectType - type of the ad shown to user. The options are: 1 - regular free ads added by users; 2 - highlighted regular (owners have to pay fixed price to highlight them and stick to the top for some period of time); 3 - contextual ads (owners have to pay per visitor's click).

HistCTR - some naive history-based estimation of click-through rate for contextual ads, calculated when the ad is showed. For non-contextual ads this field equals NULL.

IsClick - 1 if there was a click on this ad. Otherwise 0. For non-contextual ads this field equals NULL.

Can be joined with SearchInfo on Search Id, AdsInfo on AdID

**VisitsStream.tsv, PhoneRequestsStream.tsv**

These are samples of users' visits to non-contextual ad landing pages and the corresponding phone request (if one occurred). Each ad's landing page shows the hidden seller's phone number. To be able to contact the seller, the user needs to click the request phone button, which indicates a high level of interest and is in "PhoneRequestsStream"

Both can be joined with UserID on UserInfo, AdID on AdsInfo

PhoneRequestsStream has phonerequestdate instead of viewdate

Events after impression

Can be joined to impressions for conversion analysis

-Bronze layer staging table

-Silver - Joining with tables and enriching the data

Merge the tables using UserID and AdID

**testsearchstream** is not used as trainsearchStream has almost same data along with isclick.

**trainsearchStream\_staging, trainsearchstream\_silver, csv\_extract\_marker, staging\_extract\_marker** are the tables I created for bronze & silver.

**SampleSubmission** and **SampleSubmission**\_HistCTR has some prediction data for click through rate of contextual ads and not used.

**Silver Layer**

Old query for loading to silver

963 rows – takes 45 secs

After indexing, takes 0.87 sec

SELECT stg.id, stg.searchid, stg.adid, stg.position, stg.objecttype, stg.histctr, stg.isclick,

a.params, a.title, a.price,a."LocationID", a."CategoryID",

si.ipid, si.userid, si.searchquery, si.searchparams

FROM trainsearchstream\_staging stg

LEFT JOIN AdsInfo a ON stg.adid = a.adid

LEFT JOIN SearchInfo si on (stg.searchid = si.searchid AND a."LocationID" = si."LocationID"

and a."CategoryID" = si."CategoryID")

Indexes done:

CREATE INDEX idx\_stg\_adid ON trainsearchstream\_staging(adid);

CREATE INDEX idx\_stg\_searchid ON trainsearchstream\_staging(searchid);

CREATE INDEX idx\_ads\_adid ON AdsInfo(adid); //35+ sec

-- composite index to help with join filter

CREATE INDEX idx\_ads\_loc\_cat ON AdsInfo("LocationID", "CategoryID"); //44 sec

CREATE INDEX idx\_si\_searchid ON SearchInfo(searchid); // 1 min 26 sec

-- composite index to help with join filter

CREATE INDEX idx\_si\_loc\_cat ON SearchInfo("LocationID", "CategoryID"); // 1 min 50 sec

**New query for loading to silver**

SELECT stg.id, stg.searchid, stg.adid, stg.position, stg.objecttype, stg.histctr, stg.isclick,

a.params, a.title, a.price,a."LocationID", a."CategoryID",

si.ipid, si.userid, si.searchquery, si.searchparams,si."SearchDate",

c."Level" as CategoryLevel, c."ParentCategoryID" as ParentCategory, c."SubcategoryID" as SubCategory, l."Level" as Locationlevel, l."RegionID", l."CityID"

FROM trainsearchstream\_staging stg

LEFT JOIN AdsInfo a ON stg.adid = a.adid

LEFT JOIN SearchInfo si on stg.searchid = si.searchid

LEFT JOIN "Category" c ON a."CategoryID" = c."CategoryID"

LEFT JOIN "Location" l ON a."LocationID" = l."LocationID"

**Gold layer Analysis**

1. **Ad Performance Analytics**

* **CTR (Click Through Rate) by ad type**
  + CTR = clicks / impressions - Helps identify which ads (highlighted, contextual, free) perform better.
* **Top ads by clicks / conversions (phone requests / visits)**
* **Revenue contribution by ad type**
  + For contextual ads: Revenue = clicks × CPC
  + For highlighted ads: Revenue = flat highlight fee

**Ad performance – Daily CTR and Conversions by adtype**

*Provides click count, impressions count, ctr, distinct Visits count, distinct phone requests – for Ad, Ad type, params, price, title, viewDate*

create INDEX idx\_searchinfo\_searchid ON searchInfo(searchid); // 2 min 25 sec

create INDEX idx\_phone\_adid ON "PhoneRequestsStream"("AdID"); // 16.2 sec

create INDEX idx\_phone\_userid ON "PhoneRequestsStream"("UserID"); //11.2 sec

WITH ad\_ctr AS (

SELECT

ts.adid,

ts.userid,

ts.SearchDate::date AS day,

COUNT(\*) AS impressions,

COUNT(\*) FILTER (WHERE ts.isclick = 1) AS clicks,

(COUNT(\*) FILTER (WHERE ts.isclick = 1)::float / NULLIF(COUNT(\*), 0)) AS ctr

FROM trainsearchstream\_silver ts

GROUP BY ts.adid, ts.userid, day

),

phone\_requests AS (

SELECT

pr."AdID",

pr."UserID",

pr."PhoneRequestDate"::date AS day,

COUNT(\*) AS phone\_requests

FROM "PhoneRequestsStream" pr

GROUP BY pr."AdID",pr."UserID", day

)

SELECT

c.day,

c.adid,

c.userid,

c.impressions,

c.clicks,

c.ctr,

COALESCE(pr.phone\_requests, 0) AS phone\_requests

FROM ad\_ctr c

LEFT JOIN phone\_requests pr

ON c.adid = pr."AdID"

AND c.userid = pr."UserID"

AND c.day = pr.day

ORDER BY c.day, c.adid, c.userid;

*Query // 8.2 sec*

Top Ads by clicks / conversions

SELECT

ts.adid,

ts.title,

ts.ad\_type,

COUNT(ts.adid) AS impressions,

SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) AS clicks,

SUM(CASE WHEN pr."AdID" IS NOT NULL OR vs."UserID" IS NOT NULL THEN 1 ELSE 0 END) AS conversions,

ROUND(SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END)::numeric / NULLIF(COUNT(ts.adid),0), 4) AS ctr,

ROUND(SUM(CASE WHEN pr."AdID" IS NOT NULL OR vs."UserID" IS NOT NULL THEN 1 ELSE 0 END)::numeric / NULLIF(COUNT(ts.adid),0), 4) AS conversion\_rate

FROM trainsearchstream\_silver ts

LEFT JOIN "PhoneRequestsStream" pr ON ts.userid = pr."UserID" AND ts.adid = pr."AdID"

LEFT JOIN "VisitsStream" vs ON ts.userid = vs."UserID"

GROUP BY ts.adid, ts.title, ts.ad\_type

ORDER BY clicks DESC;

Revenue Contribution by type

*Assumption: consider a “conversion” generates revenue equal to price*

SELECT

ts.ad\_type,

COUNT(ts.adid) AS impressions,

SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) AS clicks,

SUM(CASE WHEN pr."AdID" IS NOT NULL OR vs."UserID" IS NOT NULL THEN ts.price ELSE 0 END) AS revenue,

SUM(CASE WHEN pr."AdID" IS NOT NULL OR vs."UserID" IS NOT NULL THEN ts.price ELSE 0 END)::numeric / NULLIF(SUM(ts.price),0) AS revenue\_share

FROM trainsearchstream\_silver ts

LEFT JOIN "PhoneRequestsStream" pr ON ts.userid = pr."UserID" AND ts.adid = pr."AdID"

LEFT JOIN "VisitsStream" vs ON ts.userid = vs."UserID"

GROUP BY ts.ad\_type

ORDER BY revenue DESC;

1. **User Behavior Insights**

* **User interest profiling**
  + Aggregate categories, locations, and ad types searched by a user.
  + Example: “User123 is mostly searching for real estate in Mumbai.”
* **Engagement metrics**
  + Avg. impressions per search, avg. CTR per user, repeat visits.
* **Conversion funnel**
  + Search → Impression → Click → Visit → Phone Request.

**User Interest Profile – Category & Location preferences**

*CREATE VIEW gold\_user\_interest\_profile AS*

SELECT

u."UserID",

COUNT(DISTINCT s.searchid) AS total\_searches,

COUNT(\*) FILTER (WHERE s.isclick = 1) AS total\_clicks,

ARRAY\_AGG(DISTINCT s.categoryid) AS preferred\_categories,

ARRAY\_AGG(DISTINCT s.regionid) AS preferred\_locations,

AVG(CASE WHEN s.isclick = 1 THEN 1 ELSE 0 END)::float AS avg\_ctr

FROM trainsearchstream\_silver s

JOIN "UserInfo" u ON s.userid = u."UserID"

GROUP BY u."UserID";

**Engagement metrics -** Avg. impressions per search, avg. CTR per user, repeat visits

create index idx\_user\_id ON trainsearchstream\_silver(userid);

create index idx\_visits\_user\_id ON "VisitsStream"("UserID") //3 min 41 sec

WITH user\_stats AS (

SELECT

ts.userid,

COUNT(ts.adid) AS impressions,

SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) AS clicks,

COUNT(DISTINCT ts.searchid) AS searches,

COUNT(DISTINCT vs."UserID" || '-' || vs."AdID" || '-' || vs."ViewDate") AS visits

FROM trainsearchstream\_silver ts

LEFT JOIN "VisitsStream" vs ON ts.userid = vs."UserID"

GROUP BY ts.userid

)

SELECT

AVG(impressions::numeric / NULLIF(searches,0)) AS avg\_impressions\_per\_search,

AVG(clicks::numeric / NULLIF(impressions,0)) AS avg\_ctr\_per\_user,

AVG(visits) AS avg\_repeat\_visits\_per\_user

FROM user\_stats;

Even after indexing ~ 56.7 sec

Conversion Funnel - Search → Impression → Click → Visit → Phone Request

SELECT

COUNT(DISTINCT ts.searchid) AS searches,

COUNT(ts.adid) AS impressions,

SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) AS clicks,

COUNT(DISTINCT vs."UserID" || '-' || vs."ViewDate") AS visits,

COUNT(DISTINCT pr."UserID" || '-' || pr."AdID" || '-' || pr."PhoneRequestDate") AS phone\_requests,

ROUND(SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END)::numeric / NULLIF(COUNT(ts.adid),0), 4) AS ctr,

ROUND(COUNT(DISTINCT vs."UserID" || '-' || vs."ViewDate")::numeric / NULLIF(SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END),0), 4) AS visit\_rate\_from\_clicks,

ROUND(COUNT(DISTINCT pr."UserID" || '-' || pr."AdID" || '-' || pr."PhoneRequestDate")::numeric / NULLIF(COUNT(DISTINCT vs."UserID" || '-' || vs."ViewDate"),0), 4) AS phone\_request\_rate\_from\_visits

FROM trainsearchstream\_silver ts

LEFT JOIN "VisitsStream" vs ON ts.userid = vs."UserID"

LEFT JOIN "PhoneRequestsStream" pr ON ts.userid = pr."UserID" AND ts.adid = pr."AdID";

1. **Search & Market Trends**

* **Trending categories & locations**
  + Which categories are seeing more searches over time (e.g., “Cars up 15% MoM”).
* **Search demand vs ad supply**
  + Searches in “Category X” vs ads available in the same category.
* **Seasonality**
  + Ads CTR or search frequency by time of day, day of week, or month.

**Search Trends – Category \* Location \* Time**

*CREATE TABLE gold\_search\_trends AS*

SELECT

s.SearchDate::date AS day,

s.categoryid,

s.regionid,

COUNT(DISTINCT s.searchid) AS total\_searches,

COUNT(\*) FILTER (WHERE s.isclick = 1) AS total\_clicks,

(COUNT(\*) FILTER (WHERE s.isclick = 1))::float / NULLIF(COUNT(\*), 0) AS ctr

FROM trainsearchstream\_silver s

GROUP BY s.SearchDate::date, s.categoryid, s.regionid;

Search Demand Vs Ad Supply

*search\_to\_ad\_ratio highlights whether demand is outpacing supply (high ratio = undersupply)*

WITH search\_demand AS (

SELECT

ts.categoryid,

COUNT(\*) AS total\_searches

FROM trainsearchstream\_silver ts

GROUP BY ts.categoryid

),

ad\_supply AS (

SELECT

ads."CategoryID",

COUNT(\*) AS total\_ads

FROM AdsInfo ads

GROUP BY ads."CategoryID"

)

SELECT

sd.categoryid,

sd.total\_searches,

COALESCE(asup.total\_ads, 0) AS total\_ads,

ROUND(sd.total\_searches::numeric / NULLIF(asup.total\_ads, 0), 2) AS search\_to\_ad\_ratio

FROM search\_demand sd

LEFT JOIN ad\_supply asup ON sd.categoryid = asup."CategoryID"

ORDER BY sd.total\_searches DESC;

Seasonality – Search frequency & CTR by hour, weekday, month

*Tracks* ***when demand peaks*** *(searches) and whether ads perform differently across* ***time buckets***

SELECT

DATE\_PART('hour', ts.SearchDate::timestamp) AS hour\_of\_day,

TO\_CHAR(ts.SearchDate::timestamp, 'Day') AS day\_of\_week,

DATE\_PART('month', ts.SearchDate::timestamp) AS month,

COUNT(\*) AS total\_searches,

SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) AS total\_clicks,

ROUND(SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END)::numeric / NULLIF(COUNT(\*), 0), 4) AS ctr

FROM trainsearchstream\_silver ts

GROUP BY hour\_of\_day, day\_of\_week, month

ORDER BY month, day\_of\_week, hour\_of\_day;

1. **Ad Quality & Pricing Insights**

* **Impact of price on CTR**
  + Average CTR grouped by ad price ranges.
* **Effect of ad type**
  + Compare CTR across free, highlighted, contextual.
* **High CTR ads with low conversions**
  + Ads with strong engagement but poor conversion (indicating possible fraud or misleading content).

**Ad Quality Vs Price – Impact of Price on CTR**

SELECT

CASE

WHEN s.price < 1000 THEN 'Low'

WHEN s.price BETWEEN 1000 AND 10000 THEN 'Medium'

ELSE 'High'

END AS price\_band,

s.price,

s.ad\_type,

COUNT(\*) AS impressions,

COUNT(\*) FILTER (WHERE s.isclick = 1) AS clicks,

(COUNT(\*) FILTER (WHERE s.isclick = 1))::float / NULLIF(COUNT(\*), 0) AS ctr

FROM trainsearchstream\_silver s

GROUP BY price\_band, s.price, s.ad\_type

ORDER BY s.ad\_type;

Effect of Ad type – Compare CTR across Free, Highlighted, Contextual

SELECT

ts.ad\_type,

COUNT(ts.\*) AS impressions,

SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) AS clicks,

ROUND(SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END)::numeric \* 100.0 / NULLIF(COUNT(ts.\*), 0), 2) AS ctr\_percent

FROM trainsearchstream\_silver ts

GROUP BY ts.ad\_type

ORDER BY ctr\_percent DESC;

High CTR but low Conversions

*Ads that get clicked often, but* ***don’t lead to visits/phone requests***

WITH ad\_stats AS (

SELECT

ts.adid,

COUNT(\*) AS impressions,

SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) AS clicks,

SUM(CASE WHEN vs."UserID" IS NOT NULL OR pr."AdID" IS NOT NULL THEN 1 ELSE 0 END) AS conversions

FROM trainsearchstream\_silver ts

LEFT JOIN "VisitsStream" vs ON ts.userid = vs."UserID"

LEFT JOIN "PhoneRequestsStream" pr ON ts.userid = pr."UserID" AND ts.adid = pr."AdID"

GROUP BY ts.adid

)

SELECT

adid,

impressions,

clicks,

conversions,

ROUND(clicks::numeric \* 100.0 / NULLIF(impressions, 0), 2) AS ctr\_percent,

ROUND(conversions::numeric \* 100.0 / NULLIF(clicks, 0), 2) AS conversion\_rate\_percent

FROM ad\_stats

WHERE clicks > 100 -- filter for meaningful traffic

AND conversions = 0

ORDER BY ctr\_percent DESC;

Ads with strong engagement but poor conversion

*Possible fraud/misleading: many clicks, almost no conversions.*

WITH ad\_metrics AS (

SELECT

ts.adid,

COUNT(\*) AS impressions,

SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) AS clicks,

SUM(CASE WHEN vs."UserID" IS NOT NULL OR pr."AdID" IS NOT NULL THEN 1 ELSE 0 END) AS conversions

FROM trainsearchstream\_silver ts

LEFT JOIN "VisitsStream" vs ON ts.userid = vs."UserID"

LEFT JOIN "PhoneRequestsStream" pr ON ts.userid = pr."UserID" AND ts.adid = pr."AdID"

GROUP BY ts.adid

)

SELECT

adid,

impressions,

clicks,

conversions,

ROUND(clicks::numeric \* 100.0 / NULLIF(impressions, 0), 2) AS ctr\_percent,

ROUND(conversions::numeric \* 100.0 / NULLIF(clicks, 0), 2) AS conversion\_rate\_percent,

CASE

WHEN clicks > 500 AND conversions = 0 THEN 'Potential Fraud'

WHEN clicks > 100 AND conversions::numeric / NULLIF(clicks,0) < 0.01 THEN 'Suspiciously Low Conversion'

ELSE 'Normal'

END AS anomaly\_flag

FROM ad\_metrics

ORDER BY ctr\_percent DESC;

1. **Location & Category Drilldowns**

* **Geo-level performance**
  + **CTR, conversion rate, and impressions at city / region / country level.**
* **Category hierarchy analysis**
  + **Electronics >** Mobile Phones > Smartphones → CTR, conversions, demand trend.

**Geo level performance**

CTR, conversion rate, impressions aggregated by city, region, country.

Took 48 sec

After indexing: 0.751 sec

CREATE INDEX idx\_pr\_user\_ad ON "PhoneRequestsStream"( "UserID", "AdID" ); //14 sec

CREATE INDEX idx\_vs\_user ON "VisitsStream"( "UserID","AdID" ); //5 min 50 sec

CREATE INDEX idx\_tss\_region\_city ON trainsearchstream\_silver(regionid, cityid); //0.051 s

SELECT

ts.regionid AS region,

ts.cityid AS city,

COUNT(\*) AS impressions,

SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) AS clicks,

ROUND(SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END)::numeric \* 100.0 / NULLIF(COUNT(\*), 0), 2) AS ctr\_percent,

SUM(CASE WHEN pr."AdID" IS NOT NULL OR vs."UserID" IS NOT NULL THEN 1 ELSE 0 END) AS conversions,

ROUND(SUM(CASE WHEN pr."AdID" IS NOT NULL OR vs."UserID" IS NOT NULL THEN 1 ELSE 0 END)::numeric \* 100.0 / NULLIF(COUNT(\*), 0), 2) AS conversion\_rate\_percent

FROM trainsearchstream\_silver ts

LEFT JOIN "PhoneRequestsStream" pr ON ts.userid = pr."UserID" AND ts.adid = pr."AdID"

LEFT JOIN "VisitsStream" vs ON ts.userid = vs."UserID" AND ts.adid = vs."AdID"

GROUP BY ts.regionid, ts.cityid

ORDER BY impressions DESC;

**Category hierarchy analysis**

**Assumption**

Level = 1 → Top-level category (e.g., Electronics)

Level = 2 → Sub-category (e.g., Mobile Phones)

Level = 3 → Leaf category (e.g., Smartphones)

CTR, conversions, trend across the hierarchy (Electronics → Mobile Phones → Smartphones).

SELECT

ts.categorylevel,

ts.categoryid,

ts.parentcategory,

ts.subcategory,

COUNT(\*) AS impressions,

SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) AS clicks,

ROUND(SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END)::numeric \* 100.0 / NULLIF(COUNT(\*), 0), 2) AS ctr\_percent,

SUM(CASE WHEN pr."AdID" IS NOT NULL OR vs."UserID" IS NOT NULL THEN 1 ELSE 0 END) AS conversions,

ROUND(SUM(CASE WHEN pr."AdID" IS NOT NULL OR vs."UserID" IS NOT NULL THEN 1 ELSE 0 END)::numeric \* 100.0 / NULLIF(COUNT(\*), 0), 2) AS conversion\_rate\_percent,

DATE\_TRUNC('month', ts.SearchDate) AS month,

COUNT(DISTINCT ts.userid) AS unique\_users

FROM trainsearchstream\_silver ts

LEFT JOIN "PhoneRequestsStream" pr ON ts.userid = pr."UserID" AND ts.adid = pr."AdID"

LEFT JOIN "VisitsStream" vs ON ts.userid = vs."UserID" AND ts.adid = vs."AdID"

WHERE ts.categorylevel <= 3 -- drill down to Smartphones

GROUP BY categorylevel, categoryid, parentcategory, subcategory, DATE\_TRUNC('month', ts.SearchDate)

ORDER BY month, impressions DESC;

1. **Fraud & Anomaly Detection (ML features)**

* **Unusual activity detection**
  + Users with abnormally high CTR / clicks in short time.
* **Ad anomaly**
  + Ads with high impressions but **0 clicks** (possible poor quality).
* **User session features**
  + Avg. time between search and click
  + # of different categories searched.

*Detect users with unusually high CTR (e.g., > 80%) or too many clicks in a short time window (say 1 hour) - Flags potential* ***bots, click farms, or fraudulent users.***

Users with abnormally high CTR or click bursts

SELECT

ts.userid,

COUNT(\*) AS impressions,

SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) AS clicks,

ROUND(SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END)::numeric \* 100.0 / NULLIF(COUNT(\*), 0), 2) AS ctr\_percent,

MIN(ts.SearchDate) AS first\_event,

MAX(ts.SearchDate) AS last\_event,

EXTRACT(EPOCH FROM (MAX(ts.SearchDate) - MIN(ts.SearchDate))) / 3600 AS active\_minutes

FROM trainsearchstream\_silver ts

WHERE ts.userid IS NOT NULL

GROUP BY ts.userid

HAVING

COUNT(\*) > 50 -- at least 50 impressions

AND (

ROUND(SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END)::numeric \* 100.0 / NULLIF(COUNT(\*), 0), 2) > 80

OR SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) > 30

)

ORDER BY ctr\_percent DESC, clicks DESC;

Ads with lots of impressions but 0 clicks

SELECT

ts.adid,

ts.title,

ts.categoryid,

ts.locationid,

COUNT(\*) AS impressions,

SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) AS clicks

FROM trainsearchstream\_silver ts

GROUP BY ts.adid, ts.title, ts.categoryid, ts.locationid

HAVING COUNT(\*) > 100 AND SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) = 0

User Session features - Avg. time between search and click

SELECT

ts.userid,

ROUND(AVG(EXTRACT(EPOCH FROM (ts.SearchDate - ts.SearchDate))), 2) AS avg\_seconds\_to\_click

FROM trainsearchstream\_silver ts

WHERE ts.isclick = 1

GROUP BY ts.userid

ORDER BY avg\_seconds\_to\_click;

Number of distinct categories searched per user

SELECT

ts.userid,

COUNT(DISTINCT ts.categoryid) AS distinct\_categories\_searched

FROM trainsearchstream\_silver ts

GROUP BY ts.userid

ORDER BY distinct\_categories\_searched DESC;

1. **ML Feature Store (Gold for Modelling)**

For training ML models (e.g., CTR prediction, recommendation engine):

* **User-level features**
  + Avg CTR, most searched categories, location preference.
* **Ad-level features**
  + Price, ad type, past CTR, category, location.
* **Search-level features**
  + Query text embedding, time of search, category hierarchy.
* **Interaction-level features**
  + Position on page, historic CTR at that slot.

**User Level features**

Average CTR, most searched categories, location preference per user

-- Step 1: count clicks per category per user

WITH user\_cat\_counts AS (

SELECT

ts.userid,

ts.categoryid,

COUNT(\*) AS cnt

FROM trainsearchstream\_silver ts

GROUP BY ts.userid, ts.categoryid

),

-- Step 2: rank categories per user

ranked\_cats AS (

SELECT

userid,

categoryid,

cnt,

ROW\_NUMBER() OVER (PARTITION BY userid ORDER BY cnt DESC) AS rn

FROM user\_cat\_counts

)

-- Step 3: pick top 3 categories per user

SELECT

userid,

ARRAY\_AGG(categoryid ORDER BY rn) AS top\_categories

FROM ranked\_cats

WHERE rn <= 3

GROUP BY userid;

**Ad Level features**

Price, ad type, past CTR, category, location

SELECT

ts.adid,

ts.price,

ts.ad\_type,

ts.categoryid,

ts.locationid,

ROUND(SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END)::numeric / NULLIF(COUNT(\*), 0), 4) AS ad\_past\_ctr,

COUNT(\*) AS impressions

FROM trainsearchstream\_silver ts

GROUP BY ts.adid, ts.price, ts.ad\_type, ts.categoryid, ts.locationid;

**Search Level features**

Query embedding, time of search, category hierarchy

SELECT

ts.searchid,

si.searchquery,

EXTRACT(HOUR FROM ts.SearchDate) AS search\_hour,

EXTRACT(DOW FROM ts.SearchDate) AS day\_of\_week,

ts.categorylevel,

ts.parentcategory,

ts.subcategory

FROM trainsearchstream\_silver ts

JOIN SearchInfo si ON ts.searchid = si.searchid

**Interaction Level Features**

Position on page, historic CTR at that slot

SELECT

ts.adid,

ts.position,

COUNT(\*) AS impressions\_at\_position,

SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END) AS clicks\_at\_position,

ROUND(SUM(CASE WHEN ts.isclick = 1 THEN 1 ELSE 0 END)::numeric / NULLIF(COUNT(\*), 0), 4) AS ctr\_at\_position

FROM trainsearchstream\_silver ts

GROUP BY ts.adid, ts.position;

**CTR Prediction**

SELECT

s.id,

s.searchid,

s.adid,

s.userid,

s.position,

s.histctr,

s.isclick,

s.price,

s.ad\_type,

s.categoryid,

s.regionid,

u."UserAgentID",

u."UserDeviceID",

EXTRACT(DOW FROM s.SearchDate) AS day\_of\_week,

EXTRACT(HOUR FROM s.SearchDate) AS hour\_of\_day

FROM trainsearchstream\_silver s

JOIN "UserInfo" u ON s.userid = u."UserID"

**Analysis on Whether table or View is required in gold layer for each business use case scenario**

|  |  |  |  |
| --- | --- | --- | --- |
| **Grouping** | **Sub Group** | **Recommended Gold storage** | **Reason** |
| Ad Performance Analytics |  |  |  |
|  | CTR (Click Through Rate) by ad type / category / region  *Helps identify which ads (highlighted, contextual, free) perform better.* | Table  Gold\_ctr\_adperf | Heavy aggregation, joins with phonerequestsstream.  Expensive to compute each time |
|  | Top ads by clicks / conversions (phone requests / visits) | Table  gold\_top\_ads | “ |
|  | Revenue contribution by ad type  *For contextual ads: Revenue = clicks × CPC*  *For highlighted ads: Revenue = flat highlight fee* | Table  gold\_adtype\_revenue | “ |
| User Behavior Insights |  |  |  |
|  | User interest profiling - Aggregate categories, locations, and ad types searched by a user.  *Example: “User123 is mostly searching for real estate in Mumbai.”* | View  Gold\_user\_profiling | Join with UserInfo only which has less rows |
|  | Engagement metrics - Avg. impressions per search, avg. CTR per user, repeat visits. | Table  gold\_engagement\_metrics | Heavy aggregations/ expensive joins with VisitsStream  Also needed indexing on userid on VisitsStream |
|  | Conversion funnel: Search → Impression → Click → Visit → Phone Request | Table  gold\_conversion\_funnel | Heavy joins.  VisitsStream and PhoneRequestsStream |
| Search & Market Trends |  |  |  |
|  | Trending categories & locations  *Which categories are seeing more searches over time (e.g., “Cars up 15% MoM”).* | Table  Gold\_search\_trends | Heavy aggregation with searchInfo  Expensive to compute each time |
|  | Search demand vs ad supply  *Searches in “Category X” vs ads available in the same category.* | Table  gold\_search\_vs\_supply | Expensive subquery joins |
|  | Seasonality - Ads CTR or search frequency by time of day, day of week, or month. | View  gold\_seasonality | Less expensive query |
| Ad Quality & Pricing Insights |  |  |  |
|  | Impact of price on CTR  *Average CTR grouped by ad price ranges.* | View  Gold\_avgctr\_by\_price | Single table query |
|  | Effect of ad type  *Compare CTR across free, highlighted, contextual* | View  gold\_ctr\_by\_adtype | Single table query  (returning summarized single row) |
|  | High CTR ads with low conversions  Ads with strong engagement but poor conversion (indicating possible fraud or misleading content) | Tables  gold\_highctr\_lowconv  gold\_suspicious\_ads | Heavy joins with PhoneRequestsstream and VisitsStream |
| Location & Category Drilldowns |  |  |  |
|  | Geo-level performance - CTR, conversion rate, and impressions at city / region / country level. | Table  Gold\_geo\_perf | Heavy aggregation with AdsInfo |
|  | Category hierarchy analysis - Electronics > Mobile Phones > Smartphones → CTR, conversions, demand trend | Table  Gold\_cat\_heirarchy | Costly joins / aggregation with Phonerequestsstream, searchInfo |
| Fraud & Anomaly Detection (ML features) |  |  |  |
|  | Unusual activity detection - Users with abnormally high CTR / clicks in short time | View  Table is historical tracking is needed, going with view  Gold\_unusual\_activity | Can be done on the fly in single table |
|  | Ad anomaly - Ads with high impressions but 0 clicks (possible poor quality) | View  Gold\_ad\_anomaly | “ |
|  | User session features - Avg. time between search and click, # of different categories searched | View  Gold\_user\_session\_anomaly  Gold\_User\_categories\_searched | “ |
| ML Feature Store |  |  |  |
|  | User-level features - Avg CTR, most searched categories, location preference | Table  gold\_user\_mlfeatures | Expensive Subqueries |
|  | Ad-level features - Price, ad type, past CTR, category, location | View  gold\_ad\_mlfeatures | Single table summary |
|  | Search-level features - Query text embedding, time of search, category hierarchy | Table  gold\_search\_mlfeatures | Expensive join with SearchInfo |
|  | Interaction-level features - Position on page, historic CTR at that slot | View  gold\_interaction\_mlfeatures | From single table |
|  | CTR Prediction | View  Gold\_ctr\_prediction | From single table |