AD CLICK THROUGH RATE PREDICTION

First: What is CTR in plain English?

CTR =
Did the user click the ad or not?

Usually stored as:

- `1` → user clicked the ad
- `0` → user did not click the ad

So CTR prediction means:

> "Given what I know about the user and the ad, will the user click this ad?"

The ML Story for CTR (Big Picture)

Think of this as a **story**, not a model.

Story:
An ad is shown → user sees it → user decides → click or no click
Your job is to **predict that decision before it happens**.

Why?

- Show better ads
- Save money
- Increase revenue

1️⃣WHY am I doing this? (Problem Thinking)

Simple meaning:

Why should anyone care about CTR prediction?

Real-world reason:

- Companies pay money **every time an ad is shown**
- If the user won't click → money wasted
- If we predict clicks → show ads only to people likely to click

Decision made using model output:

If predicted CTR is **high** → show the ad
If predicted CTR is **low** → don't show / show another ad

If model is wrong:

- Predict click but user doesn't → wasted money
- Predict no click but user would have clicked → lost revenue

So **both mistakes matter**, but missing a real click is often worse.

2️⃣WHAT exactly am I predicting? (Target Clarity)

Target variable (Y):

Usually something like:

clicked = 1 (Yes)

clicked = 0 (No)

Is this classification or regression?

Ask:

- Is output a number like 0.73? → Regression
- Is output a label like Click / No Click? → Classification

CTR prediction is **Classification**.

Even if the model outputs probability (like 0.8), the **core task is classification**.

Real-life definition matters:

Is `clicked = 1` defined as:

- Any click?
- Click within 10 seconds?
- Click after scrolling?

Different definitions = different ML problems.

3️⃣ WHAT data do I have? (Data Thinking)

Typical CTR dataset columns:

User-related:

- age
- gender
- location
- device (mobile / desktop)
- past clicks count

Ad-related:

- ad type (video / image / text)
- category (shopping, finance, gaming)

- ad position (top, side, bottom)

Context:

- time of day
- day of week
- app or website
- internet speed

Very important question:

Was this information known BEFORE the click happened?

If yes → safe
If no → data leakage (bad)

Example of leakage:

- "time_spent_after_click" → this happens *after* clicking, so cannot be used.

4️⃣ HOW does real life create clicks? (Causal Thinking)

Ask simple "why" questions.

Example:

Why would **ad position** matter?

- Ads at top are more visible → more clicks

Why would **device type** matter?

- Mobile users scroll fast → fewer clicks
- Desktop users see more clearly → more clicks

Why would **past clicks** matter?

- If user clicked ads before, they are click-friendly

This step helps you:

- Remove useless features
- Explain results to managers
- Trust your model

5️⃣HOW should I prepare the data? (Feature Thinking)

Raw data is messy. Humans don't think in raw columns.

Instead of raw values, think behavior.

Bad feature:

total_clicks = 57

Better features:

- clicks_last_7_days
- click_rate_last_30_days
- time_since_last_click
- avg_clicks_per_day

For time:

Instead of:

timestamp = 2025-01-14 18:32

Create:

- hour_of_day
- is_weekend
- is_peak_time

You are converting **raw data → meaningful signals**.

6️⃣WHICH model should I use and WHY? (Model Thinking)

CTR data characteristics:

- Mostly 0s (few people click)
- Many categorical features
- Non-linear patterns

Simple rule:

- Need explanation → Logistic Regression
- Need better accuracy → Random Forest / XGBoost

In real ad systems:

- Logistic Regression is very common (fast + interpretable)
- Tree models handle complex interactions better

Don't chase "best model".
Choose the model that **fits the data** + **business need**.

7️⃣HOW do I know my model is good? (Evaluation Thinking)

Accuracy is misleading

Example:
If only 5% users click ads:

- Predict "no click" for everyone → 95% accuracy

- But model is useless

Better metrics:

- **Precision**: When model says “click”, how often is it right?
- **Recall**: Out of real clickers, how many did we catch?
- **AUC-ROC**: Overall ranking ability

Business thinking:

Missing a real click = lost money
Showing ad to wrong user = wasted money

Usually:

- Recall is more important than accuracy
- Balance precision and recall

8️⃣ WHEN will this model fail? (Failure Thinking)

CTR models fail when:

- User behavior changes
- New ad types appear
- Festivals / sales change patterns
- Bots start clicking ads

Ask:

- Does model rely too much on one feature?
- Will this still work next month?

This tells you **when to retrain** the model.

9️⃣ HOW to explain CTR prediction simply? (Clarity

Test)

Simple explanation to a manager:

> “We trained a model using user behavior, ad details, and context to predict whether a user will click an ad.
> It learns patterns like which users click which ads at what time.
> This helps us show ads only to users likely to click, reducing wasted spend.”

If you can say this confidently → you understand the project.

One-page CTR Thinking Checklist (Very Important)

WHY

- Save ad cost
- Increase clicks
- Better targeting

WHAT

- Target: clicked (0/1)
- Type: Classification

DATA

- User + Ad + Context
- No future information

HOW

- Engineer behavior features
- Handle imbalance
- Choose correct metric

MODEL

- Logistic Regression / Tree-based

- Interpretable vs accuracy tradeoff

EVALUATION

- Precision, Recall, AUC
- Not accuracy alone

FAILURE

- Behavior change
- New ads
- Time drift

Why this framework is PERFECT for you

You:

- Get bored easily
- Get confused when things jump too fast
- Like understanding, not memorizing

This method:

- Breaks work into mental layers
- Prevents blind coding
- Makes projects feel meaningful

CTR prediction is actually a **perfect beginner-to-intermediate ML project** because:

- Clear target
- Real business value
- Simple logic
- Scales to advanced ideas later

This explanation already *means you are thinking like an ML practitioner*, not just a student.