Group Number: 05

Student ID	Student Name	Percentage of Contribution
25250841	**Vaibhav Kapri**	**40**
25315536	**Saravanakumar Chandrasekaran**	**30**
25259466	**Obinna Ohakwe**	**30**

1. Problem Identification and Initial Dataset

In this project, we aim to build a personalized **Click-Through Rate (CTR) prediction system** in the context of online advertising. CTR prediction refers to estimating the probability that a user will click on a displayed advertisement. This task is central to modern ad ranking and recommendation systems, where platforms must select and prioritise ads that maximise user engagement and revenue.
Formally, our objective is to learn a function:
P(click = 1|user,ad,context)
which assigns a click-probability to each user–ad–context combination. This predicted probability can then be used to rank ads for each user in real time. The task therefore constitutes a **binary classification problem**, where the target variable click takes value 1 when the user clicked the ad and 0 otherwise.

1.1. Loading Initial Dataset

The first step in this project involved importing the three foundational components of the Taobao Click-Through-Rate (CTR) prediction dataset: **raw_sample**, **ad_feature**, and **user_profile**. The dataset has been sourced from Kaggle (*Kaggle, Ad Display/Click Data on Taobao.com*). Each of these datasets captures a different dimension of the recommendation ecosystem and is therefore essential for building an integrated modelling framework.

The **raw_sample** file, containing 26.55 million impression-level observations, represents the core behavioural log on which CTR modelling is based. It includes user identifiers, timestamps, page identifiers (pid), and the outcome variables clk and nonclk. Initial inspection (*cf. Notebook 1.1*) showed that the dataset is entirely numeric except for the pid field, which is categorical. This early understanding was important because it signalled that significant feature expansion would be required later to capture behavioural and handling class imbalanced contextual patterns beyond the raw click information.

The **ad_feature** file, containing approximately 846,000 unique ads, provides the metadata describing each ad impression—specifically category identifiers (cate_id), campaign identifiers (campaign_id), advertiser (customer), brand, and price. The presence of both categorical and continuous fields in this dataset has important implications: categorical fields must later be encoded consistently, and continuous fields such as price require careful distributional analysis due to the presence of extreme outliers in Taobao data. Understanding these structure-level details at this stage ensures that later modelling steps are robust to inconsistencies between ad-level metadata and logged impressions.

The **user_profile** dataset, comprising 1.06 million users, contributes demographic and behavioural segmentation attributes. These include gender, age group, consumption power (pvalue_level), shopping depth, occupation, and new-user classification. Because these attributes are heavily used as targeting signals in real ad platforms, they are expected to have predictive value in CTR modelling. The datatype inspection (*cf. Notebook 1.1*) also revealed the presence of floating values in certain ordinal fields such as pvalue_level and new_user_class_level, which directly informed later data cleaning steps, including missing-value imputation and type standardisation.
This early documentation provides the foundation for the merging and alignment steps that follow, and sets expectations for the challenges to be addressed later—such as inconsistent encodings, missing values, and large-scale memory handling. The insights gained here shape the remainder of the project by highlighting the need for strategic preprocessing, efficient data merging, and careful feature engineering to extract meaningful patterns from the raw logs.

1.2. Merging the Raw Datasets

The three source datasets—raw_sample, ad_feature, and user_profile—were merged to construct a unified impression-level dataset suitable for downstream modelling. The merge was carried out in two stages.
First, the raw_sample table was joined with ad_feature on the shared key adgroup_id, ensuring that every impression record was augmented with the corresponding advertisement metadata such as category, campaign, customer, brand, and price. This step establishes the foundational link between user interactions and ad characteristics, which are central to any CTR prediction task.

Next, the user column in raw_sample was renamed to userid to harmonise naming across tables. Following this, the intermediate dataframe was merged with user_profile on userid, adding demographic and behavioural attributes including age level, shopping level, occupation, gender code, and consumption power segments. This integration is crucial because CTR behaviour is known to vary significantly across user groups (*Zhou et al., 2018*), and omitting these variables would restrict the model's ability to capture personalised patterns.

The resulting merged dataframe (*cf. Notebook 1.2*) contains advertisement-related fields, impression logs, and user-level demographic information in a single row structure. The preview of the merged dataset reveals that the joins were successful, with no evidence of structural duplication. However, the presence of missing values in several demographic fields (e.g., pvalue_level, new_user_class_level) highlights potential challenges that will need to be addressed during initial data preparation. These missing values may arise either because certain users did not have associated profile information, or because some demographic variables were sparsely populated in the source data.

This merged dataset forms the foundation for all subsequent analysis, cleaning, feature engineering, and modelling. Its structure ensures that impressions can be analysed holistically—combining ad metadata, temporal features, and user characteristics—which is essential for developing a robust CTR prediction model. The implications for later stages include the need for a systematic approach to handling missing values and careful feature engineering to maximise predictive value while preventing data leakage.

1.3. Creating the Target Variable and Time-Based Features

To prepare the merged dataset for modelling, this stage focused on constructing a unified binary target variable and extracting temporal information from the raw timestamp field (*cf. Notebook 1.3*). The raw Taobao logs encode impressions using two separate indicators—clk (1 if clicked, else 0) and nonclk (1 if not clicked, else 0). Since each row corresponds to a single impression, only one of these two fields can take the value 1. To simplify downstream modelling, a single binary variable click was created using click = 1 − nonclk, after which both original columns were dropped. This transformation produces a clean and interpretable dependent variable aligned with conventional CTR-prediction practice. (*cf. Notebook 1.3*)

The next operation involved converting the UNIX-style time_stamp column into Python datetime format. This conversion enables the extraction of structured temporal features known to influence user engagement behaviour. From the processed timestamp, the attributes hour, minute, second, day, weekday, and date were derived. These engineered variables allow the later model to capture diurnal patterns (e.g., users tending to click more during evening hours) *(Wang et al., 2024; Zhou et al., 2020*), weekday–weekend differences, and general temporal trends. An inspection of the converted timestamps reveals that the dataset spans a continuous eight-day window, from **5 May 2017 16:00:00** to **13 May 2017 15:59:46**. This relatively short but uninterrupted period has two important implications for the project.

- First, it allows the construction of leakage-free chronological train–test splits later in Section 3, where the final day (13 May) can be reserved as unseen test data to simulate real production deployment.
- Second, the dense temporal coverage provides a strong foundation for subsequent sequence-based features (e.g., recency and user impression indices), which depend on accurately ordered event logs.

Overall, the transformations carried out in this step establish a coherent target variable, enrich the dataset with temporal context, and prepare the data for deeper behavioural and sequence feature engineering in later stages of the project.

1.4. Handling Missing Values

A critical early stage of the data preparation process involved identifying and resolving missing values within the merged dataset. Missing data, if left untreated, can lead to biased descriptive statistics, unstable feature engineering outputs, and significant degradation in model performance. Therefore, a systematic audit and imputation process was carried out (*cf. Notebook 1.4*).

Audit of Missing Values

An initial NaN inspection revealed that most fields—such as userid, adgroup_id, cate_id, campaign_id, customer, price, and all timestamp-derived components—contained **no missing values**, confirming the structural integrity of the merged data. (*cf. Notebook 1.4*)
However, three key attributes exhibited gaps:

- **pvalue_level** — a user-level ordinal consumption grade
- **brand** — an ad-level categorical metadata field
- **new_user_class_level** — an ordinal city-tier classification

These missing entries required tailored treatments because their semantic interpretation differs across user and ad dimensions.

Imputation Strategy and Rationale

1. **pvalue_level (Ordinal User Consumption Grade):** Missing values were filled using the **mode**, which is appropriate for ordinal categorical variables *(Chawla et al., 2002).* This approach preserves class distribution (low, medium, high) without introducing artificial numeric distortion.
2. **brand (Categorical Ad Identifier):** Since brand is categorical and missing values do not imply numeric meaning, they were replaced with a sentinel value **-1**, representing an "unknown brand". This ensures:
   - the model can isolate patterns associated with missing brand metadata,
   - information loss is minimised without fabricating artificial relationships.
3. **new_user_class_level (Ordinal City Level):** As an ordinal variable similar to pvalue_level, missing entries were replaced using the **mode** (Mode imputation is appropriate for ordinal categorical variables (*Chawla et al., 2002*), preserving the underlying population distribution.
4. **Column Name Correction:** The field new_user_class_level initially contained a trailing whitespace. Renaming this column prevented referencing errors, especially during grouping and aggregation steps used later for feature engineering.

Implications for Later Stages

- Clean input ensures that **feature engineering**—particularly CTR-based aggregates and sequence features—will not include misleading NaN-based calculations.
- The mode-based imputations maintain **distributional integrity**, which is essential for fair model learning.
- The sentinel brand = -1 allows the model to treat previously missing brand metadata as an **interpretable and learnable pattern**, rather than noise.
- Consistency across all attributes prepares the dataset for stable scaling, encoding, and eventual model training.

This step ensures that the dataset is structurally sound and ready for downstream feature engineering and model experimentation.

1.5. Fixing Data Types

Following the treatment of missing values, the next essential step in preparing the Taobao advertising dataset was the correction of data types. Although the raw merged dataset contained all necessary fields, several variables were stored using inappropriate numeric formats that did not reflect their conceptual meaning. Left untreated, these mismatches could mislead both exploratory analysis and downstream modelling. (*cf. Notebook 1.5*)

First, all **identifier-level variables**—including userid, adgroup_id, cate_id, campaign_id, customer, brand, and pid—were explicitly cast to the category data type. These attributes represent discrete entities rather than quantities with arithmetic meaning. Treating them as integers could incorrectly signal ordinality or magnitude, potentially biasing algorithms that are sensitive to numeric scale. Recasting them as categorical ensures that models learn based on membership rather than magnitude, and also optimises memory usage because pandas stores categorical values more efficiently than raw integers.

Second, the set of **demographic attributes** (cms_segid, cms_group_id, final_gender_code, age_level, pvalue_level, shopping_level, occupation, new_user_class_level) were also converted to categorical types. These variables are encoded numerically in the data source but represent discrete socio-demographic classes. Categorising them prevents inappropriate statistical operations (e.g., averaging ages or gender codes) and ensures correct behaviour during grouping, feature engineering, and model encoding steps later in the pipeline.

By standardising these data types at this early stage, the project establishes a clean and semantically correct foundation for the subsequent exploratory analysis (Section 2), feature engineering (Section 4), and machine learning modelling. Additionally, saving the output of this step as a standalone checkpoint (resultOfStep1.csv) supports reproducibility and provides a recovery point should later experimentation require revisiting earlier transformations.

2. Description of Dataset / Exploratory Data Analysis (EDA)

Once the raw datasets were merged, cleaned, and structurally prepared in Section 1, the next phase involved a systematic exploration of the dataset to understand its behavioural, temporal, and demographic characteristics. This exploratory analysis serves as a diagnostic layer between data preparation and modelling, enabling the identification of patterns, biases, anomalies, and structural relationships that directly influence the choice of appropriate preprocessing strategies and machine learning techniques.

The Taobao dataset is inherently high-dimensional and event-driven, containing tens of millions of user–ad impressions accompanied by metadata on users, ads, campaigns, brands, and demographic attributes. Such scale and heterogeneity necessitate a structured EDA approach—beginning with a review of variable types and distributional properties, followed by an assessment of missingness, behavioural trends (e.g., CTR variation across demographic segments), and the presence of outliers or extreme values in critical features such as price.
Through this process, EDA provides three essential benefits for the remainder of the project:

1. **Understanding Relationships:** Identifying how user attributes, ad properties, and temporal patterns interact with click behaviour, guiding early hypotheses for feature engineering.
2. **Detecting Data Issues:** Revealing inconsistencies, skewed distributions, or structural outliers that could distort model learning if left unaddressed.
3. **Informing Modelling Decisions:** Providing empirical basis for algorithm selection, feature transformations, and balancing strategies used later in iterative modelling pipeline.

Overall, this section establishes a comprehensive understanding of the working dataset, ensuring that subsequent modelling is both statistically sound and aligned with the behavioural realities of the Taobao platform.

2.1. Attribute Definitions (Meaning and Value Types)

The merged dataset contains a combination of user-level demographics, ad-level metadata, contextual variables, and the binary click-through indicator. Before proceeding with feature engineering or modelling, it is essential to establish a clear understanding of the semantic meaning and data type of each attribute. This ensures that subsequent preprocessing decisions (e.g., encoding, scaling, handling of outliers, and construction of derived features) are aligned with the underlying nature of the variables rather than their raw numeric representation.

The definitions below are based on the official Taobao Display Advertising Challenge documentation, supplemented by an inspection of the actual unique values present in our dataset. These attributes represent the complete set of variables available immediately after merging the raw impression logs (raw_sample), ad metadata (ad_feature), and user demographics (user_profile) in Section 1. (*cf. Notebook 2.1*)

Table 2.1 — User Identifiers & Static Attributes

Attribute	**Type**	**Value Range / Categories**	**Meaning**
userid	Nominal (Identifier)	Unique IDs	Unique user identifier; no inherent order.

final_gender_code	Categorical (Binary)	1 = Male, 2 = Female	User’s gender.
age_level	Ordinal categorical	0 = Unknown, 1 = <18, 2 = 18–24, 3 = 25–29, 4 = 30–34, 5 = 35–39, 6 = 40+	User’s age bracket.
pvalue_level	Ordinal categorical	1 = Low, 2 = Medium, 3 = High	Spending/consumption level.
shopping_level	Ordinal categorical	1 = Shallow, 2 = Moderate, 3 = Deep	Browsing/purchasing depth.
occupation	Categorical (Binary)	0 = Non-student, 1 = College student	Student status.
new_user_class_level	Ordinal categorical	1–4	User’s city-tier classification (higher = lower tier).
cms_segid	Nominal categorical	Various segment IDs	Micro-segment identifier.
cms_group_id	Nominal categorical	Various group IDs	Higher-level CMS segmentation group.

Table 2.2 — Ad Attributes

Attribute	**Type**	**Value Range / Categories**	**Meaning**
adgroup_id	Nominal (Identifier)	Unique IDs	Unique identifier for an ad group; corresponds to a specific advertised item.
cate_id	Nominal categorical	Various category IDs	Product category of the advertised item.
campaign_id	Nominal (Identifier)	Unique campaign IDs	Marketing campaign under which the ad is grouped.
customer	Nominal (Identifier)	Advertiser IDs	Represents the advertiser (merchant).
brand	Nominal categorical	Brand IDs (including -1 for unknown)	Brand associated with the advertised product.
price	Numerical (float)	Wide range, includes outliers	Price of the advertised item displayed in the ad.

Table 2.3 — Contextual Attributes

Attribute	**Type**	**Value / Categories**	**Meaning**
time_stamp	Numerical (int64, UNIX timestamp)	Seconds since epoch	Exact timestamp of impression event.
pid	Nominal categorical	e.g., *430539_1007*, *430548_1007*	Scenario/page type where the ad was displayed (e.g., homepage, search results).
hour	Numerical (int)	0–23	Hour extracted from timestamp.
day	Numerical (int)	1–31	Day of month.
weekday	Numerical (int)	0–6	Day of week (0 = Monday).
date	Date	YYYY-MM-DD	Calendar date of impression.
minute	Numerical (int)	0–59	Minute extracted from timestamp.
second	Numerical (int)	0–59	Second extracted from timestamp.

Table 2.4 — Target Variable

Attribute	**Type**	**Values**	**Meaning**
click	Binary (0/1)	0 = No click, 1 = Click	Indicates whether the displayed ad was clicked by the user.

2.2. Data Integrity and Consistency Checks

Before conducting any exploratory analysis, it was essential to validate the structural integrity and reliability of the merged dataset. This step ensured that subsequent modelling and feature engineering would be grounded in clean, logically consistent data. (*cf. Notebook 2.2*)

First, a duplicate analysis revealed **zero fully duplicated rows**, indicating that no impression logs were inadvertently repeated during the merge process. However, when checking for repeated impression keys (userid–adgroup_id–date), the dataset contained **378,330 duplicate combinations**. This pattern is expected rather than problematic: users typically see the same advertisement multiple times within a day, so repeated user–ad–day entries reflect *true behavioural exposure* rather than data noise. As such, these observations were retained.

Second, the validity of key identifier fields was assessed. All major ID columns (userid, adgroup_id, cate_id, campaign_id, customer, brand, pid) contained **zero missing or zero values**, confirming that the merge pipeline preserved identifier integrity. This reduces the likelihood of silent join errors, null key failures, or misaligned relational mappings between impression logs, ad metadata, and user profiles.

Third, categorical variables were inspected for value sanity. The distribution of **gender**, **age levels**, and **shopping level** strictly adhered to the expected documented ranges of the Taobao dataset. No unexpected category codes or structural anomalies were identified, confirming that demographic attributes were both complete and semantically valid.

Finally, hierarchical consistency between ad components was verified. For every adgroup_id, the associated cate_id, campaign_id, and brand values exhibited **zero inconsistencies**, meaning that each adgroup was linked to exactly one category, one campaign, and one brand. This behaviour is expected in a stable advertising system where adgroups serve as atomic units. The absence of conflicts confirms that joins were performed correctly and no merging artefacts distorted the ad hierarchy.

Overall, the integrity checks validate that the dataset is clean, consistent, and structurally coherent. This provides a robust foundation for the subsequent Exploratory Data Analysis and feature engineering stages, where any patterns uncovered can be interpreted with confidence rather than skepticism about data quality issues.

2.3. Target Variable: Click vs Non-click Distribution

In the our dataset, the target click indicates whether a user clicked an advertisement (1) or did not click (0). The distribution observed from the merged dataset is highly imbalanced: **94.87% of impressions result in no click**, whereas **only 5.13% result in a click**. This level of imbalance is typical of real-world display advertising logs (*Zhang et al., 2020; Xiong et al., 2020*), where the majority of users scroll past ads without interacting. (*cf. Notebook 2.3*) Such imbalance has several implications for subsequent modelling stages.

First, traditional accuracy-based evaluation becomes misleading because a naïve classifier predicting “always no click” would already achieve nearly 95% accuracy. Therefore, relying exclusively on accuracy is inappropriate for model selection. Instead, we must emphasise metrics that reflect minority-class behaviour—such as precision, recall, F1-score, and ROC-AUC—which will be systematically incorporated in Section 4 during iterative evaluation.

Second, the imbalance guides model design decisions. Algorithms such as logistic regression, decision trees, or random forests may naturally favour the dominant class unless appropriately regularised or reweighted. Later experiments may benefit from techniques such as **class balancing**, **threshold tuning**, or **probability calibration**, which are especially important in CTR tasks where the marginal utility of correctly predicting a rare click is much higher than correctly predicting many non-clicks. Moreover, the imbalance encourages a stronger focus on informative behavioural features in Section 3, which can help models distinguish subtle patterns that signal user interest.

In summary, the examination of the target variable reveals a heavily skewed distribution that directly shapes evaluation strategy, model selection, and the importance of robust feature engineering. This insight forms a foundational reference point for all subsequent stages of the project.

2.4. CTR by Attributes

Understanding how click-through rates vary across different user, ad, and contextual attributes is essential for identifying meaningful behavioural patterns in the data. Since CTR prediction is inherently tied to user intent and ad relevance, analysing attribute-level CTR provides early insight into which variables may carry predictive signal. This exploratory phase helps validate assumptions from marketing theory (e.g., user demographics affect engagement), reveals potential feature importance, and highlights whether any attributes show negligible variance and may be excluded later. (*cf. Notebook 2.4*)

Moreover, CTR patterns across groups assist in motivating later feature engineering choices. For example, strong differences in gender-, age-, or category-level CTR can justify the inclusion of behavioural aggregations or interaction terms. These findings will also inform model diagnostics in Section 4, helping interpret why certain models perform better based on the underlying structure of the data.

2.4.1. CTR by Gender

CTR variation across genders was analysed to understand whether demographic segmentation influences ad engagement behaviour (*cf. Notebook 2.4.1*). The dataset contains two gender groups—**Male** (final_gender_code = 1) and **Female** (final_gender_code = 2). The computed click-through rates indicate a subtle but consistent difference: male users exhibit a CTR of **4.83%**, whereas female users show a higher CTR of **5.24%**.

Although this difference is modest, it is non-trivial in the context of large-scale advertising platforms, where even a 0.4–0.5% shift can represent a significantly higher probability of clicking. This suggests that gender could act as a weak but meaningful predictor in the CTR model. Importantly, both groups remain highly imbalanced due to the naturally low click rate in display advertising, but the *relative difference* across categories offers insight into behavioural heterogeneity.

From an analytical perspective, identifying such CTR asymmetries early helps us anticipate how demographic attributes contribute to model learning. While gender alone is unlikely to be a strong stand-alone predictor, its interaction with other attributes (e.g., category preferences or time-of-day effects) may become influential. This preliminary result therefore supports retaining the gender variable for subsequent feature engineering and model training steps.

2.4.2. CTR by Age Level

Click-through behaviour shows measurable variation across age groups, as revealed by the analysis in *cf. Notebook 2.4.2*. After mapping Taobao's encoded age brackets to interpretable labels, CTR was computed by averaging the binary *click* indicator within each age segment. This allowed us to compare engagement patterns across demographic cohorts.

The results indicate a **mild U-shaped pattern** in age-related responsiveness. Users aged **≤18** recorded a CTR of **5.52%**, one of the highest among all groups, suggesting that younger users may be more receptive or exploratory in their browsing behaviour. Engagement dips slightly in the large mid-age bracket (19–34), where CTR values range between **4.95% and 5.17%**, indicating relatively lower inclination to click on ads, possibly due to more targeted browsing or reduced novelty-seeking. Users aged **≥40** show the **highest CTR at 5.62%**, mirroring the trend seen in the youngest group and suggesting increased likelihood of ad interaction, perhaps linked to higher purchasing intent or availability of disposable income.

The finding that both the youngest and oldest user groups exhibit higher engagement has methodological implications for later modelling stages. It suggests that **age_level** may hold predictive value for CTR modelling, warranting its inclusion in feature engineering and interaction-based experiments. Additionally, understanding demographic engagement differences supports better segmentation strategies and highlights the importance of demographic-aware feature construction in Section 4.

Overall, the analysis confirms that age is a **relevant behavioural dimension** in CTR prediction, with variations substantial enough to matter both statistically and practically for downstream model development.

2.4.3. CTR by Consumption Grade

The consumption value level (pvalue_level) is an ordinal attribute that reflects a user's spending ability, broadly segmented into low, medium, and high purchasing power. Understanding its relationship with CTR helps determine whether higher-value users engage more actively with ads—an insight that may later influence both model design and feature engineering.

As shown in *cf. Notebook 2.4.3*, the analysis included mapping the numerical codes to user-friendly category labels and computing the mean CTR for each grade. The results revealed a notable irregularity: several combinations of pvalue_level and the derived label appeared with **NaN CTR values**. This indicates that, for those particular value–label pairs, there were **no recorded impressions**, resulting in undefined CTRs rather than genuine zero engagement. This is most likely due to the presence of redundant rows created by the merge operation between pvalue_level and its label column.

Focusing on the valid rows, the meaningful CTR values were:

- **Low Consumption Users**: CTR ≈ **0.05146**
- **Medium Consumption Users**: CTR ≈ **0.05148**
- **High Consumption Users**: CTR ≈ **0.04719**

Two observations emerge from these results:

1. **Low and Medium consumption users behave nearly identically**, both around a 5.15% CTR. This suggests that mid-range spending power does not independently predict higher ad engagement.
2. **High consumption users demonstrate a lower CTR**, approximately 4.7%, indicating that users with higher purchasing capacity are **less likely to click** on general display ads.

This pattern aligns with behaviours observed in e-commerce ecosystems, where higher-value users tend to browse more selectively and click less frequently on broad-targeted advertisements. The implication for future modelling is significant: spending power may **not** be a strong positive predictor of click probability unless combined with interaction terms or personalised preferences (e.g., user-category affinity).

Given these patterns, pvalue_level will be retained as a feature but treated cautiously during modelling. We may later examine its importance through feature selection methods (e.g., Chi-Square) to validate whether it contributes meaningfully to predictive performance.

2.4.4. CTR by Shopping Depth

The analysis in *cf. Notebook 2.4.4* evaluates how a user's browsing depth—captured by the variable shopping_level—correlates with advertisement engagement. Shopping depth is an ordinal measure describing how extensively a user interacts with Taobao's platform, ranging from *Shallow* (level 1) to *Deep* (level 3). It reflects underlying behavioural intensity, including browsing duration, purchase frequency, and breadth of category exploration.

The computed CTR patterns reveal a subtle but meaningful gradient. *Shallow* users exhibit the highest click-through rate (5.398%), while *Moderate* (5.172%) and *Deep* (5.113%) users show slightly lower propensities. This behaviour contrasts with an intuitive expectation that deeper, more engaged users might click more often. Instead, the results suggest that deep users may be more experienced or selective, possibly engaging in more targeted browsing and thus showing greater resistance to display ads.

This insight contributes significantly to later modelling decisions. It implies that shopping depth is unlikely to behave as a monotonically increasing predictor; rather, its relationship with CTR appears nuanced and may require capturing non-linear patterns during feature engineering (e.g., binning or one-hot encoding). Furthermore, the difference—although small—could influence how the model captures behaviour across heterogeneous user types. These observations guide the choice of feature transformations and help set realistic expectations for model performance when demographic and behavioural variables show only marginal separation between click and non-click groups.

2.4.5. CTR by Occupation

In this subsection, we examined whether a user's occupational status influences their likelihood of clicking on an advertisement (*cf. Notebook 2.4.5*). The dataset contains a binary classification—**student** versus **non-student**—captured through the occupation variable. Since student users are often associated with higher digital engagement and exploratory browsing behaviour, evaluating this attribute allowed us to determine whether it meaningfully contributes to click propensity.

The results show that the difference between both groups is **negligible**. Students have a CTR of **0.05152**, while non-students have a nearly identical CTR of **0.05131**. The extremely small gap (~0.00021) indicates that occupational status, at least in this coarse two-level representation, does **not** meaningfully affect click behaviour. Consequently, occupation appears to be a **weak standalone predictor** for CTR.

Nevertheless, this attribute still carries relevance for downstream modelling. First, although occupation alone provides little information, it may interact with other variables—such as age level, shopping depth, or user-level behavioural features—in ways detectable by **non-linear learning algorithms**. Second, including occupational information preserves demographic completeness in the modelling dataset. Lastly, clarifying that this attribute contributes minimal signal helps us avoid over-weighting it during feature selection and guides later efforts in engineered features and interaction terms.

Overall, our analysis suggests that **occupation does not materially influence CTR**, but we retain it as a potential **interaction feature** for more expressive models introduced in Section 4.

2.4.6. CTR by City Tier

To assess whether user location plays a meaningful role in ad responsiveness, we analysed CTR patterns across the four city-tier levels defined in the dataset. These tiers reflect the socio-economic and market maturity of the user's city, with **Tier-1** representing highly urbanised regions and **Tier-4+** representing comparatively lower-tier cities.
The results (*cf. Notebook 2.4.6*) reveal an extremely narrow CTR band across all tiers:

City Tier	CTR
Tier-1	0.050427
Tier-2	0.051715
Tier-3	0.050602
Tier-4+	0.051089

We observe that **Tier-2 cities show the slightly highest CTR**, while **Tier-1 and Tier-3 cities show very similar CTR levels**, and **Tier-4+ sits between Tier-2 and Tier-3**. However, the absolute differences are small—less than **0.002** across all tiers—indicating that **city tier alone is not a strong discriminator of click behaviour**.

This finding helps shape our subsequent modelling approach. While the attribute may still contribute marginally in combination with others, its standalone predictive value appears limited. Thus, in later feature-importance analyses, we anticipate that the model may not assign high weight to this feature unless it interacts meaningfully with high-value user-behavioural or contextual attributes.

Overall, the CTR pattern across city tiers confirms that **geographic economic tier does not substantially segment user click propensity** in this dataset. This insight allows us to prioritise more influential demographic, behavioural, and ad-level features when designing feature-engineering experiments.

2.4.7. CTR by PID (Scenario)

In this subsection, we analyse whether the **ad serving scenario**, captured through the PID attribute, affects user engagement. As shown in *cf. Notebook 2.4.7,* the dataset contains two PID categories—430539_1007 and 430548_1007—which we label as **Scenario A** and **Scenario B**, respectively, to support interpretability.

The CTR analysis reveals a measurable difference between the two environments: **Scenario A records a CTR of 0.0535**, while **Scenario B shows a lower CTR of 0.0499**. Although the absolute difference is modest, the pattern indicates that the **context or placement where an ad is displayed influences user behaviour**. This finding is logically consistent with real-world advertising: certain page layouts or browsing flows are more conducive to clicks than others.

The implication for our modelling pipeline is twofold. First, PID must be retained as a categorical predictor because it encapsulates behavioural variation that is not explained by user or ad-level features alone. Second, the scenario-specific differences suggest that **contextual signals matter in CTR prediction**, and future feature engineering can consider deeper combinations (e.g., *scenario × hour*, *scenario × category*) if model performance plateaus. This observation also supports the need for non-linear models during later iterations in Section 4, where interactions across contextual and behavioural attributes may be captured more effectively.

2.5. Price Distribution & Outlier Analysis

The price attribute exhibits substantial variability, making it one of the most heterogeneous numerical variables in the dataset. As shown in the descriptive summary (*cf. Notebook 2.5*), while the median price is modest (≈ **168**), the distribution contains a long right-tail extending to an extreme maximum value of **100,000,000**. This results in a standard deviation (~ **130,943**) that is orders of magnitude larger than the mean (~ **749**), indicating severe positive skewness.

The linear-scale histogram (*cf. Notebook 2.5*) collapses almost the entire distribution into a single bar, visually confirming extreme skew. After applying a log(1+price) transformation, the distribution becomes approximately bell-shaped, revealing that **most ads fall within a narrow mid-price range**, with only a tiny proportion extending into the tens of thousands or millions.

The boxplot (*cf. Notebook 2.5*) further highlights the presence of **very large outliers**, with points scattered far beyond the whiskers. Based on numerical inspection, **530 impressions** (~0.0021% of the dataset) have prices above **1,000,000**, and importantly, **46** of these extreme-price ads recorded at least one click.

Interpretation of price behaviour
The presence of extremely high price values (e.g., 99,999,999) suggests one of the following scenarios:

1. **System encoding placeholders** (e.g., “missing price”, “premium/negotiated price”, “unadvertised price”).
2. **Outlier product categories** such as luxury real estate, automobiles, or rare items.
3. **Data quality issues** arising from upstream logging systems.

Despite their rarity, some of these extreme-price observations do produce clicks, indicating that they may correspond to genuine impressions rather than artefacts.

We examined three potential strategies:

Strategy	Pros	Cons
Remove outliers	Simplifies modelling, prevents instability	Risk of discarding meaningful rare events; removes *clicked* observations
Cap (winsorize)	Avoids distortion of scaling models; retains observations	Introduces artificial ceiling; may compress useful variation
Keep as-is	Preserves natural distribution; avoids information loss	Causes numerical instability in models requiring scaling; large influence on distance-based models

Given that outliers include **46 clicked impressions**, outright removal would risk deleting informative rare click behaviour. However, keeping the raw extreme values can hinder model training, especially for:

- **Logistic Regression** (unstable gradients)
- **KNN, SVM** (distance-based sensitivity)
- **Tree ensembles** (may over-partition extreme regions)

Decision

We therefore **retain the raw price values for EDA**, and in later modelling stages:

- **Create price_bin features** for non-linear patterns.
- Maintain the **original uncapped price** for tree-based models that can exploit high-variance signals without scaling issues.

This dual-treatment ensures minimal information loss while protecting model stability. The strong skewness suggests price will likely benefit from transformations or binning in feature engineering stages.

2.6. Pearson Correlation Analysis (Numerical Features)

In this section, we extended our exploratory analysis by examining the linear relationships among the numerical and ordinal variables included in the dataset. Using the correlation heatmap generated in *cf. Notebook 2.6* , we aimed to assess the extent of multicollinearity and uncover any strong dependencies that could influence feature engineering or model performance.

The results demonstrate that **most numerical variables exhibit extremely weak correlations with one another**, with coefficients clustering very close to zero. For instance, price shows negligible correlation with all demographic and segmentation attributes (|r| < 0.003). This suggests that ad pricing operates independently of user-level segmentation variables, reinforcing the assumption that Taobao's pricing is primarily ad-driven rather than user-driven.

The strongest relationship is the well-known dependency between **CMS segmentation variables**:

- cms_segid ↔ cms_group_id (r ≈ 0.40)

This is expected because cms_group_id is a broader grouping of the finer-grained cms_segid. Such relationships reflect structural hierarchy rather than redundant information. However, both variables may partially overlap in predictive signal, which is a factor we will consider during feature importance analysis in later modelling stages.

A more notable insight emerges from the strong negative correlation between **cms_group_id** and **final_gender_code** (r = –0.92). Since final_gender_code is only a binary male/female indicator, the magnitude of this correlation likely arises from how Taobao internally assigns CMS groups based on demographic strata. It emphasizes that CMS segmentation embeds demographic dimensions implicitly. This is useful for feature interpretation but also indicates potential redundancy: demographic signals might already be captured within CMS variables.

Similarly, moderate correlations appear between age_level and CMS groupings (0.24–0.49), as well as between spending power (pvalue_level) and age or segmentation. These correlations indicate that older users tend to fall into different CMS segments and higher pvalue brackets, reflecting real consumption patterns in digital commerce. Importantly, we observe **no pairs of variables approaching problematic multicollinearity thresholds** (typically |r| > 0.80), except the structural pair between gender and cms_group_id. Since this relationship stems from Taobao's internal segmentation design rather than accidental redundancy, we have not removed either variable at this stage.

Implications for Subsequent Stages

- The low correlation across most features supports our plan to retain a wide range of attributes for modelling, as each contributes potentially independent signal.
- The hierarchical dependency between CMS variables justifies later use of tree-based models (e.g., Random Forest, XGBoost), which handle correlated features more robustly than linear models.
- The strong gender–CMS correlation suggests caution when interpreting model coefficients in linear models, as demographic effects may be partially inseparable from CMS segmentation.

- No immediate dimensionality reduction (e.g., PCA) is required at this stage since multicollinearity risk remains minimal.

Overall, this correlation assessment reassures us that the feature space is structurally diverse and suitable for building a CTR prediction model without heavy preprocessing.

2.7. Spearman Correlation Analysis (Ordinal Features)

(cf. Notebook 2.7)
In this subsection, we analyse the **monotonic associations** among all ordinal attributes using the **Spearman rank correlation coefficient**. Unlike Pearson's correlation, which measures linear dependence, Spearman's ρ captures ordered but potentially non-linear relationships, making it well suited for variables such as age groups, spending power levels, occupation status, shopping depth, and city tiers.

The correlation matrix reveals several structurally significant patterns. The strongest relationship is observed between **cms_group_id** and **final_gender_code** (ρ ≈ –0.78), followed by **cms_group_id** and **age_level** (ρ ≈ 0.68). These high-magnitude correlations indicate that Taobao's internal CMS segmentation effectively captures demographic ordering, especially around gender and age. While this provides behavioural insight into the platform's segmentation strategy, it also signals potential **multicollinearity risks**, which we will need to address during model preparation.

Moderate associations, such as between **age_level** and **pvalue_level** (ρ ≈ 0.17), highlight that users in older age brackets generally tend to exhibit higher consumption power. Similarly, the positive correlations of **shopping_level** with **cms_segid** and **new_user_class_level** suggest clustering of deeper shoppers within specific CMS segments or particular city tiers.

At the same time, several variable pairs demonstrate minimal monotonic correlation—e.g., **occupation** vs. **new_user_class_level**, or **gender** vs. **city tier**—indicating that these behavioural and demographic characteristics operate largely independently.

The implications for subsequent modelling are meaningful. High correlations among certain ordinal features may require **regularisation**, **feature selection**, or **combination strategies** to prevent model instability. Conversely, low correlations among others reinforce their potential unique contribution to predictive performance. As we move into feature engineering, this matrix will guide decisions around encoding, dimensionality reduction, and the prioritisation of attributes that capture the most behavioural signal without redundancy.

3. Initial Data Preparation

3.1. Feature Engineering

The feature-engineering stage transforms the merged dataset into a modelling-ready structure capable of capturing behavioural, temporal and contextual signals relevant for click-through-rate prediction. All transformations were performed chronologically and in a leakage-free manner to preserve modelling integrity (*cf. Notebook 3.1*).

The first set of transformations focused on enriching the temporal context of each impression. Using the hour and weekday attributes, we derived part_of_day (night, morning, afternoon, evening) and is_weekend, allowing the model to recognise differences in user engagement patterns across daily and weekly cycles. These contextual signals matter because platform traffic and user intent vary significantly by time of day, and modelling such variation enables more responsive CTR estimation. (*Wang et al., 2024; Zhou et al., 2020*)

We then computed several user-level aggregate features. For each userid, we calculated total impressions, total clicks, overall user CTR, and the user's historical exposure to average ad price. These attributes characterise broad behavioural tendencies—such as how frequently a user clicks or whether they interact more with high- or low-value ads—thereby giving the model insight into user heterogeneity beyond raw demographic variables (*cf. Notebook 3.1: user_stats & user_price*).

To incorporate sequential behaviour, we constructed a datetime field from date and time components, chronologically sorted impressions by user, and computed user_impression_index and secs_since_last_impression. These recency-based features reflect the spacing and order of exposures, which is important because the likelihood of clicking can depend on how recently or frequently the user was shown ads. For example, too many impressions in a short period may lead to fatigue, whereas spaced impressions often signal renewed intent.

Next, ad-level performance priors were computed at multiple hierarchies: ad_ctr, cate_ctr, brand_ctr, campaign_ctr, and customer_ctr. These values summarise the global attractiveness of ads, product categories, brands, campaigns and advertisers. In advertising platforms, certain adgroups consistently perform stronger than others, and providing the model with these priors helps it contextualise each impression beyond user-side behaviour. (*Richardson et al., 2024*)

Given the strong right-skew and extreme outliers in price (with values reaching 99,999,999), we assigned ads to price_bin categories (low, mid, high, extreme). This keeps the signal while reducing the impact of outliers, which would otherwise distort linear and tree-based models.

We also encoded the page scenario identifier pid through pid_enc after label-encoding. This ensures that contextual differences between page placements (e.g., different sections of Taobao's interface) can be learned by the model.

A personalised feature capturing user–category affinity was added by computing user_cate_ctr, representing each user's historical click-through rate for a specific product category. This type of collaborative filtering signal is known to greatly enhance behavioural models by embedding users' past interest patterns.

Finally, to ensure leakage-free training, we derived strictly historical ("past-only") behavioural statistics using sequential expanding windows: user_impressions_past, user_clicks_past, user_ctr_past, and user_avg_price_past. These values exclude the current impression by applying a shift, ensuring that the model never accesses future information. Leakage-free design is essential in CTR prediction: without it, the model would artificially inflate performance and fail to generalise in deployment. (*Kaufman et al. (2012*).

After generating all features, non-modelling columns such as pure identifiers (userid, adgroup_id), raw time fields (time_stamp, datetime, minute, second), and unencoded PID were removed to prevent unnecessary memory usage and eliminate sources of overfitting. The resulting feature set contains **37 engineered variables** and approximately **25 million records**. This dataset now provides a rich, behaviourally expressive foundation for subsequent model development.

3.2. Feature Selection and Correlation Analysis

In this section, we evaluate which engineered features are most predictive of click-through behaviour using the **Chi-square (χ²) statistical test**. The analysis helps us understand which variables contribute meaningful discriminatory power and which ones may be redundant. Because the Taobao CTR dataset is extremely high-volume and exhibits strong class imbalance, feature selection serves two purposes simultaneously: improving model learning efficiency and reducing unnecessary computational load—a critical requirement considering our later modelling workflow (*cf. Notebook 3.2*).

We begin by partitioning the dataset chronologically into **training (2017-05-05 to 2017-05-12)** and **testing (2017-05-13)** to preserve temporal integrity. This ensures that the χ² statistics are computed strictly on historical data, preventing any temporal leakage. After separating the target variable, we restrict the feature matrix to **non-object numerical columns**, because χ² requires non-negative numeric values. A small number of features (e.g., *brand*) contained negative values due to earlier transformations; therefore, we apply a constant shift to make them non-negative without altering their internal ordering—ensuring correctness of the χ² computation.

Running **SelectKBest(χ²)** on all numeric features reveals a striking pattern: a handful of behavioural and temporal variables dominate the predictive signal. Specifically, **secs_since_last_impression** emerges overwhelmingly as the strongest predictor, followed by **brand**, **price**, and **customer** identifiers. This aligns with established literature in personalised advertising, where recency and product-level familiarity strongly influence click propensity. User-level cumulative metrics such as **user_impressions**, **user_avg_price_past**, and **user_impression_index** also appear highly influential, reinforcing the value of sequential behavioural modelling introduced in our feature engineering pipeline.

The **scree plot** (*cf. Notebook 3.2, Scree Plot*) demonstrates a steep drop after the top few features, and the **cumulative predictive strength curve** shows that:

- The **top 5 features** already capture **≈95.5%** of total χ² signal.
- The **top 10 features** capture **≈99.8%**.
- Beyond **15 features**, the contribution becomes negligible.

This pattern has significant implications for downstream modelling. It tells us that a small but carefully constructed subset of behavioural, temporal, and price-related features carries nearly the entire predictive strength for CTR. In practical terms, this means that highly complex models may not need the full dimensionality of the engineered dataset, and simpler models such as logistic regression or naïve Bayes may perform competitively when supplied with the optimal feature subset. At the same time, keeping the full feature space available allows advanced models like Random Forests or Gradient Boosting to potentially exploit weaker interactions.

Overall, this feature selection exercise validates the effectiveness of our feature engineering in Section 3.1 and provides a principled basis for selecting an efficient and high-signal feature set for model training. The next steps in our modelling pipeline will use these insights to compare baseline and optimised learning algorithms under varying levels of feature richness, ensuring both methodological rigor and computational feasibility.

4. Iterative Process

4.1. Baseline Modelling Using Top-20 Features

In this stage of the iterative process we moved from exploratory analysis and feature engineering to systematic baseline modelling using the top-20 predictors selected via the Chi-square procedure (*cf. Notebook 3.2).* Our goal was to establish a robust reference point for performance before introducing more advanced models and imbalance-handling strategies.

4.1.1. Constructing Train/Test Matrices

Using the chronologically split data created earlier (train: 5–12 May; test: 13 May; *cf. Notebook 3.2*), we first restricted both train_df and test_df to the twenty highest-ranking features identified by the Chi-square scree and cumulative strength plots. These included temporal recency (secs_since_last_impression), price and brand, customer and campaign identifiers, several user-history aggregates (user_impressions, user_avg_price_past, user_impressions_past, user_impression_index, user_avg_ad_price), personalised preference (user_cate_ctr), base structural features such as cate_id, and a set of CTR-type aggregates and price_bin. We then extracted X_train and X_test as NumPy matrices and y_train, y_test as binary targets (*cf. Notebook 4.1.1*).

Because several algorithms considered later (KNN, SVM, Random Forest) are computationally heavy on tens of millions of rows, we also created smaller, stratified-style samples for these models: 500,000 rows from the training set and 100,000 rows from the test set. This design allowed us to compare model families under controlled resource constraints while still training Logistic Regression and Naive Bayes on the full dataset.

4.1.2. Feature Scaling

Before fitting the models, we applied standardisation using scikit-learn's StandardScaler (*cf. Notebook 4.1.2*). All twenty features in X_train were scaled to zero mean and unit variance, and the same transformation was applied to X_test using the fitted scaler. This step is particularly important for distance- and margin-based methods such as KNN and SVM, where unscaled variables with different units (e.g. price vs. counts) can dominate the distance metric or decision boundary. Standardisation also tends to stabilise optimisation in Logistic Regression and improves convergence behaviour.

4.1.3. Model Selection and Training Strategy

We then configured a diverse set of baseline algorithms (*cf. Notebook 4.1.3*): Logistic Regression, Gaussian Naive Bayes, K-Nearest Neighbours, Decision Tree, Support Vector Machine, and Random Forest. This set spans linear models, probabilistic classifiers, distance-based learners, and tree-based classifiers, allowing us to understand how different inductive biases respond to the engineered CTR features and severe class imbalance.

To manage computational cost fairly, we trained Logistic Regression and Naive Bayes on the full scaled train set, while KNN, SVM, and Random Forest were trained on the reduced X_train_small / X_test_small matrices. Each model produced class predictions and probability estimates on its respective test split, enabling calculation of Accuracy, Precision, Recall, F1-score and ROC-AUC, as well as confusion matrices stored for later comparison (*cf. Notebook 4.1.4* result table).

4.1.4. Baseline Performance and Interpretation

The resulting metrics show a consistent pattern. Logistic Regression and Random Forest delivered the strongest overall performance, with ROC-AUC values around 0.93 and 0.94 respectively, and balanced trade-offs between precision, recall and F1-score. These results indicate that even relatively simple linear decision boundaries, when combined with well-designed behavioural and temporal features, can capture much of the signal in this CTR prediction task, while Random Forest further exploits non-linear interactions.

Naive Bayes and Decision Tree achieved moderate performance. Naive Bayes showed relatively high recall but lower precision, reflecting its simplifying independence assumptions and a tendency to over-predict the minority class.

Decision Tree achieved one of the best recall values among the baselines but at the cost of lower ROC-AUC, consistent with unregularised trees overfitting local patterns.

KNN and SVM were the weakest performers in this configuration. KNN's recall was extremely low despite reasonable accuracy, confirming that local distance methods struggle in high-dimensional, highly imbalanced spaces. SVM effectively collapsed to predicting the majority class on the sampled data, yielding zero recall and F1-score for the positive class. This behaviour reinforced that, without careful rebalancing and hyperparameter tuning, SVM is not competitive for this large-scale CTR problem.

Implications for Subsequent Iterations

These baseline results, together with the confusion matrices and ROC-AUC values (*cf. Notebook 4.1.4*), informed several key decisions for the remainder of the project.

First, we selected **Logistic Regression** and **Random Forest** as our primary baselines against which any subsequent advanced model (e.g. gradient boosting, CatBoost) must improve.

Second, we deprioritised KNN and SVM in later stages, since their poor recall and computational demands suggested low return on additional tuning effort.

Third, the strong performance of tree-based models with engineered behavioural features justified our investment in leakage-safe user histories and ad-level CTRs in Section 3, and motivated exploring more sophisticated boosting methods that can better handle class imbalance while capturing complex interactions.
Overall, this baseline modelling stage established both a quantitative benchmark and a qualitative understanding of which model families are promising for CTR prediction on our dataset. It also provided concrete evidence that meaningful gains will likely come from more powerful tree-based ensembles and imbalance-aware training strategies, rather than from further tuning of distance-based or margin-based classifiers.

4.2. Focused Training on Random Forest

Following the baseline modelling experiments in Section 4.1, we transitioned into a more focused and systematic optimisation process centred on the Random Forest classifier. This choice was informed directly by the initial model comparison (*cf. Notebook 4.1*), where Random Forest demonstrated a strong balance between AUC performance and learning capacity under class imbalance. Section 4.2 documents the full iterative workflow we undertook—from hyperparameter tuning to threshold optimisation and training on a rebalanced dataset—while explicitly discussing how each step influenced subsequent modelling decisions.

4.2.1. Hyperparameter Search Using RandomizedSearchCV

To improve baseline Random Forest performance, we executed a hyperparameter search over a carefully constrained parameter grid (*cf. Notebook 4.2.1*). We used a **RandomizedSearchCV** approach to optimise the model on a computationally manageable subset of 500,000 training rows, focusing on **F1-score**, which is the most appropriate metric under severe CTR class imbalance. (*Fawcett, 2006*)

The search identified the following optimal configuration:

- **n_estimators = 400**
- **max_depth = 40**
- **min_samples_split = 10**
- **min_samples_leaf = 6**
- **max_features = 'sqrt'**
- **bootstrap = True**
- **class_weight = 'balanced'**

This parameter set reflects a strong bias towards generalisation under skewed classes. In particular, the min_samples_leaf=6 and min_samples_split=10 constraints help reduce overfitting in the long-tail distribution of non-click impressions.

When evaluated on the small test subset, this tuned model achieved an F1-score of **0.5531** at the default threshold (0.5), with recall reaching **0.5834**, indicating that the model was now substantially better at capturing positive (click) instances compared to the baseline.

The hyperparameter tuning step established a stronger model foundation and directly shaped our next refinement—threshold optimisation—because the imbalance still caused the model to output conservative probabilities for the positive class.

4.2.2. Threshold Tuning for F1 Optimisation

Given that Random Forests output a probability distribution, the default decision threshold of **0.50** is often suboptimal in imbalanced CTR tasks. We therefore scanned thresholds between 0.05 and 0.51 (*cf. Notebook 4.2.2*) and evaluated the resulting F1-scores.

The best performance was achieved at:

- **Optimal threshold: 0.49**
- **F1 = 0.5531**
- **Precision = 0.5258**
- **Recall = 0.5834**

This demonstrated that even slight threshold adjustments significantly influence the balance between false positives and false negatives. Threshold tuning improved the model's responsiveness to minority-class clicks, but the remaining skew in the dataset limited recall further. This motivated our move into dataset balancing.

4.2.3. Balancing the Training Data (Downsampling Majority Class)

Given that the original training data exhibited a **95:5 ratio** between non-click and click impressions (*cf. Notebook 4.2.3*), we created a **balanced training dataset** through random downsampling of the majority class.

- Original:
    - Non-click = 21.21M
    - Click = 1.15M
- After downsampling:
    - Non-click = 1.15M
    - Click = 1.15M

This rebalanced dataset offers the model equal exposure to both classes, enabling it to learn the characteristics of positive events effectively. Balancing was expected to improve recall substantially, but at the cost of precision, since Random Forests trained on heavily downsampled data often become more aggressive in predicting the minority class.

4.2.4. Retraining Random Forest on Balanced Data & Threshold Optimisation

After rebalancing, we retrained the Random Forest using the tuned hyperparameters (cf. *Notebook 4.2.4).* On the small balanced training/test split:

- **Accuracy = 0.8300**
- **Precision = 0.2180**
- **Recall = 0.9289**
- **F1 = 0.3532**
- **AUC = 0.9540**

The recall improvement was dramatic, as expected, but precision dropped sharply because the classifier now predicted many more false positives due to the now-equal class proportions.

To correct this, we performed threshold tuning again (0.05–0.90):

- **Optimal threshold = 0.83**
- **F1 = 0.5705**
- **Precision = 0.5330**
- **Recall = 0.6136**

This restored the precision-recall balance while retaining the gains from balancing. This iterative process validated the need to treat threshold selection as part of the modelling pipeline—not a post-processing convenience—because threshold tuning systematically corrects bias introduced by resampling.

4.2.5. Final Random Forest on Full Balanced Dataset

We then trained the final Random Forest on the entire balanced dataset using the best hyperparameters and applied the optimised threshold **0.83** (*cf. Notebook 4.2.5*). Performance on the full test set was:

- **Accuracy = 0.9587**
- **Precision = 0.5771**
- **Recall = 0.6678**
- **F1 = 0.6191**
- **ROC-AUC = 0.9663**

This constituted the strongest Random Forest performance achieved in all experiments, improving overall F1 by nearly **+10%** compared to the baseline and substantially strengthening recall without sacrificing too much precision. This established Random Forest as a competitive, well-balanced candidate model. It also provided a rigorous performance benchmark for subsequent ensemble models (XGBoost and CatBoost) evaluated in Section 4.3.

4.3. Experiments with Ensemble Models

While Random Forest delivered a strong tuned performance, modern CTR prediction systems increasingly rely on gradient-boosted trees due to their ability to model non-linear and high-interaction patterns (Zhou et al. 2018). Section 4.3 documents the exploratory evaluation of **XGBoost** and **CatBoost** using the same balanced training subset to ensure comparability.

4.3.1. XGBoost on Balanced Subset

XGBoost was trained on the same small balanced dataset using a conservative configuration to prevent overfitting (*cf. Notebook 4.3.1*). The model achieved:

- **Accuracy = 0.8560**
- **Precision = 0.2462**
- **Recall = 0.9129**
- **F1 = 0.3878**
- **ROC-AUC = 0.9584**

Although AUC was strong, the F1-score lags behind Random Forest. The recall pattern suggests that XGBoost, when trained on downsampled data, becomes overly sensitive to minority-class examples without sufficient precision to support robust CTR prediction. These results indicated that XGBoost may require extensive hyperparameter tuning or alternative strategies (e.g., scale_pos_weight) to match Random Forest performance on imbalanced CTR tasks.

4.3.2. CatBoost on Balanced Subset

We next evaluated CatBoost (*cf. Notebook 4.3.2*), particularly because it handles categorical splits more naturally in many classical tabular datasets (Dorogush, Ershov & Gulin 2018). Initial performance on the balanced small test set:

- **Accuracy = 0.8553**

- **Precision = 0.2458**
- **Recall = 0.9169**
- **F1 = 0.3876**
- **ROC-AUC = 0.9603**

The results were nearly identical to XGBoost, reflecting a similar precision-recall imbalance. However, CatBoost exhibited higher AUC and smoother probability calibration.
Threshold tuning identified:

- **Optimal threshold = 0.8596**
- **Best F1 ≈ 0.5957**

This suggested that CatBoost had the capacity for significant improvement if tuned properly, motivating a dedicated hyperparameter search.

4.3.3. Final Tuned CatBoost on Full Balanced Dataset

We performed a RandomizedSearchCV over CatBoost's key hyperparameters (iterations, depth, learning rate, subsample, border count) to optimise F1-score (*cf. Notebook 4.3.3*). The best configuration was:

- **iterations =** 300
- **depth =** 6
- **learning_rate =** 0.1
- **l2_leaf_reg =** 9
- **subsample =** 1.0
- **border_count =** 128

We then trained the final model on the full balanced dataset and evaluated it on the full test set using the previously identified optimal threshold.
Final performance:

- **Accuracy =** 0.9631
- **Precision =** 0.6347
- **Recall =** 0.6237
- **F1 =** 0.6291
- **ROC-AUC =** 0.9693

5. Final Model and Evaluation

This section consolidates the outcome of the iterative modelling pipeline and presents the final selected model, the evaluation strategy applied on unseen data, and the resulting predictive performance. In keeping with the project guidelines, this section integrates:
(a) the rationale behind selecting the final model,
(b) the methodology used to generate predictions on unseen instances, and
(c) evidence-based evaluation supported by graphical and statistical outputs.

5.1. Selection of the Final Model

Based on the experiments conducted across Section 4, CatBoost emerged as the strongest candidate for final deployment. The model consistently outperformed Random Forest, XGBoost, and the baseline learners across multiple criteria. Specifically:

- Its **ROC-AUC on the small test subset exceeded 0.96**, reflecting superior ranking ability.
- After balancing the training data and conducting hyperparameter tuning (*cf. Notebook 4.3.2*), CatBoost achieved **F1 ≈ 0.596** on the small balanced test set at the optimised threshold.
- When retrained on the **full balanced dataset** and evaluated on the **full unseen test set**, the tuned CatBoost model achieved:

Table 5.1 Result Matrix for CatBoost

Metric	**Final Score (Full Test)**
Accuracy	0.9631
Precision	0.6347
Recall	0.6237
F1-score	0.6291
ROC-AUC	0.9693

These results were obtained using the optimised threshold identified earlier (*cf. Notebook 4.3.3),* ensuring a balanced trade-off between precision and recall. Compared to Random Forest—which achieved F1 ≈ 0.6191—CatBoost provided a **higher F1-score, higher AUROC, and better calibration**, justifying its selection as the final model.

5.2. Generating Predictions on Unseen Data

The trained CatBoost model was used to produce probabilistic predictions on the unseen test dataset. Consistent with the earlier modelling pipeline, no additional scaling or encoding was required for the test features because:

1. All transformations (binning, CTR aggregation, sequence features) were applied **before** the train–test split;
2. The CatBoost model handles numerical feature scaling internally;
3. Using the previously fitted scaler ensures consistency with training, avoiding data leakage (*cf. Notebook 3.1*).

Predicted probabilities were subsequently thresholded using the tuned decision boundary **t = 0.8596**, selected earlier for maximising F1 (*cf. Notebook 4.3.3*). This threshold emphasises a balanced click-prediction strategy: aggressively filtering low-probability impressions while still capturing a significant proportion of true clicks.

5.3. Evidence: ROC Curve on Full Test Set

The ROC curve (*cf. Notebook 4.3.4*) provides intuitive visual evidence of the model's discriminative power. The curve displayed a rapid rise toward the upper-left quadrant, with an AUROC of **0.969**, indicating excellent class separation.

This graphical result supports earlier numerical findings and confirms that:

- the model correctly ranks clicked impressions above non-clicked impressions in the vast majority of cases;
- CatBoost maintains high stability even when exposed to the full scale of unseen test data;
- the model generalises well beyond the balanced training set, showing no sign of overfitting.

Integrating this visual evidence into the evaluation increases confidence in real-world applicability, particularly in ranking-based CTR systems such as ad-serving engines.

5.4. Interpretation of Final Results

The final scores indicate strong predictive quality despite the extremely low natural click-through rate (≈5.1%). Key interpretive points include:

1. **High AUROC (0.9693)**: Demonstrates the model's ability to order impressions correctly by click likelihood, an essential requirement in advertising platforms.
2. **Balanced Precision (0.6347) and Recall (0.6237)**: Suggests that the model is not merely predicting the majority class (non-clicks) but is successfully identifying a substantial fraction of true click events.
3. **Strong F1-score (0.6291)**: Given the severe class imbalance of the original dataset, this value reflects meaningfully improved performance compared to all baseline learners.
4. **Stable generalisation to the full test set**: The consistency between small-test and full-test performance confirms robustness of the learned patterns rather than overfitting to the balanced training subset.

5.5. Implications for CTR Prediction

The overall evaluation indicates that the tuned CatBoost model is well-suited for deployment in a CTR prediction pipeline. Practical implications include:

- **Enhanced Ad Targeting:** Higher ranking accuracy directly translates into better ad-to-user matching.
- **Improved Budget Efficiency:** Higher precision ensures fewer wasted impressions on low-probability clicks.
- **Scalability:** CatBoost can handle heterogeneous engineered features without needing heavy pre-processing.
- **Threshold Flexibility:** The ROC and predicted probability distributions allow adjusting thresholds for varying business priorities, such as recall-sensitive ad campaigns.

In sum, the final model provides a strong, evidence-backed foundation for CTR prediction in large-scale e-commerce advertising systems.

References

Kaggle (2018) Ad Display/Click Data on Taobao.com. Available at: https://www.kaggle.com/datasets/pavansanagapati/ad-displayclick-data-on-taobaocom?select=user_profile.csv

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002) 'SMOTE: synthetic minority over-sampling technique', *Journal of Artificial Intelligence Research*, 16(1), pp. 321–357. Available at: https://doi.org/10.1613/jair.953

Fawcett, T. (2006) 'An introduction to ROC analysis', *Pattern Recognition Letters*, 27(8), pp. 861–874. Available at: https://doi.org/10.1016/j.patrec.2005.10.010

Wang, R., Xue, S., Li, J., Shan, L., Guan, Z., Wu, L., Zhang, W. and Gai, K. (2024) 'Temporal interest network for user response prediction', in *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion)*. Singapore: ACM, pp. 899–908. Available at: https://doi.org/10.1145/3589335.3648340

Richardson, E., Trevizani, R., Greenbaum, J. A., Carter, H., Nielsen, M. and Peters, B. (2024) 'The receiver operating characteristic curve accurately assesses imbalanced datasets', *Scientific Reports*, 14, 11370. Available at: https://doi.org/10.1038/s41598-024-62018-7

Xiong, Z., Zhang, J., Chen, D., Zhu, G. and Zhang, W. (2020) 'Click through rate effectiveness prediction on mobile ads using extreme gradient boosting', *Computers, Materials & Continua*, 66(2), pp. 1589–1604. Available at: https://doi.org/10.32604/cmc.2020.012460

Zhang, W., Chen, M., Li, Y., Chen, J. and Hong, R. (2020) 'A new click-through rates prediction model based on deep&cross network', *Algorithms*, 13(12), 342. Available at: https://doi.org/10.3390/a13120342

Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H. and Gai, K. (2018) 'Deep interest network for click-through rate prediction', in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. London, UK: ACM, pp. 1059–1068. Available at: https://doi.org/10.1145/3219819.3219823

Kaufman S, Rosset S, Perlich C & Stitelman O (2012) ‘Leakage in data mining: Formulation, detection, and avoidance.’ *ACM Transactions on Knowledge Discovery from Data*, 6(4), pp.1–21. Available at: https://doi.org/10.1145/2382577.2382579

Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, X. & Gai, K. 2018, *Deep Interest Network for Click-Through Rate Prediction*, Proceedings of the 24th ACM SIGKDD Conference, pp. 1059–1068.

Dorogush, A.V., Ershov, V. & Gulin, A. 2018, *CatBoost: Unbiased Boosting with Categorical Features*, arXiv preprint arXiv:1810.11363, viewed [insert date], https://arxiv.org/abs/1810.11363