Database Query Assistant • Chat Transcript

Exported: 11/30/2025, 6:35:44 PM

👤 You

generate an overlayed bar chart showing the difference in relative distibutions of series group between the smote and the ads data

🤖 Assistant

The relative distributions of `series_group` show notable differences between the original `ads` data and the `synthetic_train_smote` data. The `series_group` column, which typically ranges from 2 to 8, exhibits the following patterns:

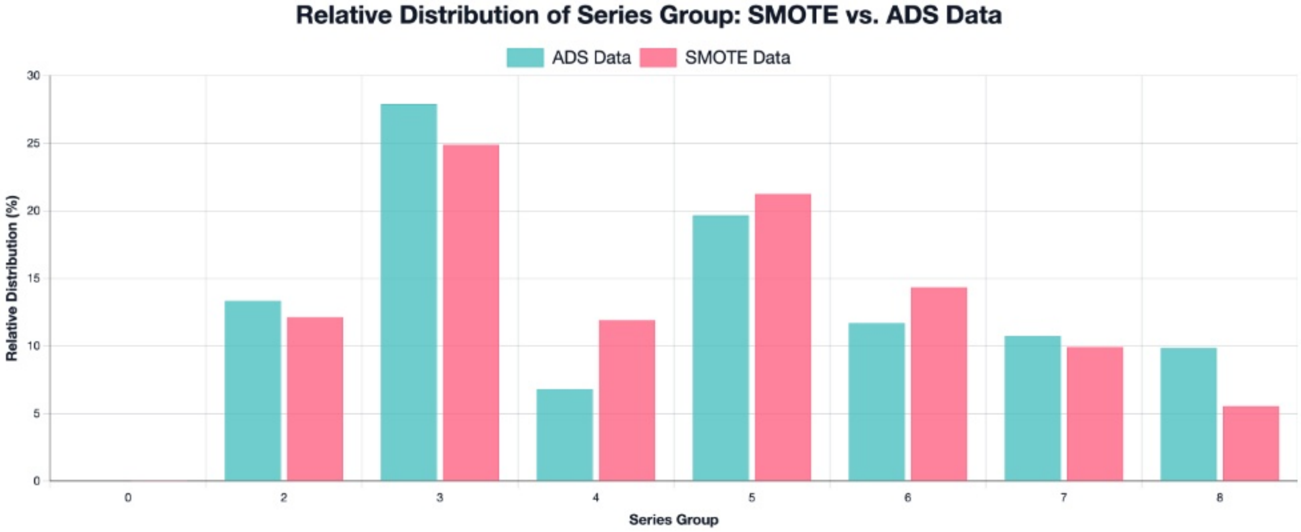
- **`series_group` 3** is the most prevalent category in both datasets, but it is more dominant in the `ads` data (27.90%) compared to the `smote` data (24.90%), a difference of 3.00 percentage points.
- **`series_group` 8** is significantly less represented in the `smote` data (5.55%) than in the `ads` data (9.85%), showing a substantial difference of 4.30 percentage points.
- Conversely, **`series_group` 4** has a much higher relative distribution in the `smote` data (11.91%) compared to the `ads` data (6.81%), a difference of 5.10 percentage points.
- **`series_group` 6** also shows a higher proportion in the `smote` data (14.33%) than in the `ads` data (11.69%), a difference of 2.64 percentage points.
- **`series_group` 5** is slightly more represented in `smote` (21.26%) than in `ads` (19.66%), a difference of 1.60 percentage points.
- **`series_group` 2** and **`series_group` 7** have smaller differences, with `ads` showing slightly higher proportions (13.34% vs 12.13% for group 2, and 10.74% vs 9.92% for group 7).
- A `series_group` value of 0 appears in the `smote` data with a negligible distribution (0.00001%), which is not present in the `ads` data.

In summary, the `smote` dataset has altered the distribution of `series_group` by increasing the relative frequency of categories like 4, 5, and 6, while decreasing the frequency of categories such as 3, 7, and 8, compared to the original `ads` dataset.

SQL Queries Executed:

```
SELECT 'smote' AS source, series_group, COUNT(*) AS count, COUNT(*) * 100.
0 /(SELECT COUNT(*) FROM synthetic_train_smote) AS relative_distribution FROM
synthetic_train_smote GROUP BY series_group UNION ALL SELECT 'ads' AS
source, series_group, COUNT(*) AS count, COUNT(*) * 100.0 / (SELECT COUNT(*)
FROM ads) AS relative_distribution FROM ads GROUP BY series_group ORDER BY
source, series_group
```

15 row(s) returned



---

source	series_group	count	relative_distribution
ads	2	1023633	13.33634
ads	3	2141706	27.90308
ads	4	522779	6.81099
ads	5	1509368	19.66471
ads	6	897124	11.68812
ads	7	824592	10.74315
ads	8	756315	9.85360
smote	0	1	0.00001
smote	2	839852	12.13045
smote	3	1723829	24.89823
smote	4	824871	11.91408
smote	5	1471812	21.25821
smote	6	992223	14.33123
smote	7	686667	9.91792

smote	8	384245	5.54987

👤 You

what is the relationship between the ads and feeds data?

🤖 Assistant

The `ads` and `feeds` data are related through the `user_id` column, specifically `ads.user_id` and `feeds.u_userId`. This relationship is explicitly shown in the `ads_feeds_joined` view, which performs a `LEFT JOIN` on these columns.
The core relationship is a cross-domain one:

- **`ads` (Target Domain):** Contains information about advertisements, user interactions with ads (like clicks, represented by the `label` column), and user profiles.
- **`feeds` (Source Domain):** Contains user behavior data and basic information about news items from the news feed domain.

The purpose of this relationship is to enrich the `ads` data with user behavior and interests from the `feeds` domain. By leveraging user activity in news feeds (e.g., `u_newsCatInterests`, `u_refreshTimes`, `u_feedLifeCycle`), the goal is to build more comprehensive user interest models. This, in turn, helps to improve the click-through rate (CTR) prediction performance for ads, addressing the sparsity of user behavior data if only the `ads` domain were used.

👤 You

Give me the top 3 users from the ads data with the highest CTR (minimum 100 ad appearances)

🤖 Assistant

The top 3 users from the `ads` data with the highest Click-Through Rate (CTR), considering only users with a minimum of 100 ad appearances, are:

1. **User ID 165362**: Achieved a CTR of **25.63%** from 41 clicks out of 160 ad appearances.
2. **User ID 140728**: Achieved a CTR of **24.29%** from 43 clicks out of 177 ad appearances.
3. **User ID 116969**: Achieved a CTR of **17.07%** from 28 clicks out of 164 ad appearances.

SQL Queries Executed:

```
SELECT user_id, SUM(label) AS total_clicks, COUNT(*) AS total_appearances,
(SUM(label) * 100.0 / COUNT(*)) AS ctr FROM ads GROUP BY user_id HAVING
COUNT(*) >= 100 ORDER BY ctr DESC LIMIT 3
```

3 row(s) returned

---

user_id	total_clicks	total_appearances	ctr
165362	41	160	25.62500
140728	43	177	24.29379
116969	28	164	17.07317