

## Moderator:

use challenge;

-- High Performing Moderator

WITH ranked\_moderators AS (

SELECT

moderator\_id,

AVG(user\_satisfaction\_score) AS avg\_satisfaction,

AVG(avg\_response\_time) AS avg\_response\_time,

SUM(chat\_sessions\_moderated) AS total\_sessions,

RANK() OVER (ORDER BY AVG(user\_satisfaction\_score)

DESC,SUM(chat\_sessions\_moderated) desc, AVG(avg\_response\_time) asc) AS rank\_high

FROM moderator\_performance

GROUP BY moderator\_id

)

SELECT

moderator\_id,

avg\_satisfaction,

avg\_response\_time,

total\_sessions

FROM ranked\_moderators

WHERE rank\_high <= 5

ORDER BY rank\_high;

--Low Performng Moderator

WITH ranked\_moderators AS (

SELECT

moderator\_id,

AVG(user\_satisfaction\_score) AS avg\_satisfaction,

AVG(avg\_response\_time) AS avg\_response\_time,

SUM(chat\_sessions\_moderated) AS total\_sessions,

RANK() OVER (ORDER BY AVG(user\_satisfaction\_score) ASC, AVG(avg\_response\_time)

desc,SUM(chat\_sessions\_moderated) asc ) AS rank\_low

FROM moderator\_performance

GROUP BY moderator\_id

)

SELECT

moderator\_id,

```
    avg_satisfaction,  
    avg_response_time,  
    total_sessions  
FROM ranked_moderators  
WHERE rank_low <= 5  
ORDER BY rank_low;
```

*-- Analyze Response Time and User Satisfaction*

```
WITH response_time_bins AS (  
    SELECT  
        moderator_id,  
        user_satisfaction_score,  
        chat_sessions_moderated,  
        avg_response_time,  
        CASE  
            WHEN avg_response_time <= 5 THEN '0-5 mins'  
            WHEN avg_response_time <= 10 THEN '5-10 mins'  
            WHEN avg_response_time <= 15 THEN '10-15 mins'  
            WHEN avg_response_time <= 20 THEN '15-20 mins'  
            ELSE '20+ mins'  
        END AS response_time_range  
    FROM moderator_performance  
)  
SELECT  
    response_time_range,  
    AVG(user_satisfaction_score) AS avg_satisfaction,  
    SUM(chat_sessions_moderated) AS total_sessions  
FROM response_time_bins  
GROUP BY response_time_range  
ORDER BY response_time_range;
```

	moderator_id	avg_satisfaction	avg_response_time	total_sessions
1	c3f76036-879a-4b96-abca-a0fe289bae01	5	13.93	460
2	df47acaf-2bf3-4bdf-b50f-ba99b8d2fa89	5	15.66	459
3	34bf3ff2-34d3-4105-966f-ada0b1332093	5	16.7	447
4	38790191-59cb-43ec-81aa-41e76ebc3cc3	5	7.08	435
5	6ed567f1-772b-4615-9ca1-5bb42ee7a9c8	5	5.01	413

  

	moderator_id	avg_satisfaction	avg_response_time	total_sessions
1	1de0004c-c52a-4f9d-b038-d89d0c0646fd	1	30.71	373
2	59c2aae8-034f-4328-a878-d940f8a503a9	1	28.97	430
3	94cd72c7-08bf-404a-9659-5545af4883ab	1	18.16	98
4	bdc6928a-7809-459e-8369-e1f53da93148	1	15.96	146
5	2b85ea0c-ec66-4cab-85cc-5128d99e2a17	1	5.35	180

  

	response_time_range	avg_satisfaction	total_sessions
1	0-5 mins	2	2855
2	10-15 mins	3	6752
3	15-20 mins	3	6110
4	20+ mins	1	803

Query executed successfully

## User Activity and Recommendation:

use challenge;

with lead\_recommndation

as

(

select u.user\_id,r.feedback\_score, u.timestamp, r.recommendation\_type,

Lead(r.recommendation\_type) over (order by u.timestamp asc) as

next\_recommendation\_type,

r.click\_through\_rate,

Lead(r.click\_through\_rate) over (order by u.timestamp asc) as next\_click from user\_activity

u

join recommendation r on u.user\_id= r.user\_id

),

get\_users as

(

select user\_id from recommendation group by user\_id having count(\*)>1

),

grouped\_podcast as

(

```

        select recommendation_type,next_recommendation_type, DATENAME(WEEKDAY,
timestamp) as day, avg(next_click-click_through_rate) as diff
        from lead_recommedation l join get_users g on l.user_id=g.user_id
        group by recommendation_type,next_recommendation_type,DATENAME(WEEKDAY,
timestamp)

)
select * from grouped_podcast
--where recommendation_type='Podcast' and next_recommendation_type='Podcast'
order by day,diff desc ;

```

	recommendation_type	next_recommendation_type	day	diff
1	Podcast	Podcast	Friday	0.365
2	Video	Video	Friday	0.29
3	Podcast	Video	Friday	0.205
4	Blog	Blog	Friday	0.2
5	Video	Podcast	Friday	-0.035
6	Blog	Podcast	Friday	-0.27
7	Video	Blog	Friday	-0.385
8	Podcast	Blog	Friday	-0.42
9	Video	Blog	Monday	0.185
10	Blog	Podcast	Monday	0.18
11	Podcast	Video	Monday	0.163333333333333
12	Blog	Blog	Monday	-0.008
13	Blog	Video	Monday	-0.0933333333333334
14	Video	Podcast	Monday	-0.105714285714286
15	Podcast	Blog	Monday	-0.18
16	Video	Video	Monday	-0.24
17	Podcast	Podcast	Monday	-0.275
18	Video	Podcast	Saturday	0.55
19	Blog	Podcast	Saturday	0.09
20	Blog	Video	Saturday	0.0833333333333333

```

with lead_recommedation
as
(
select u.user_id,r.feedback_score, u.timestamp, r.recommendation_type,
Lead(r.recommendation_type) over (order by u.timestamp asc) as
next_recommendation_type,
r.click_through_rate,
Lead(r.click_through_rate) over (order by u.timestamp asc) as next_click from user_activity
u
join recommendation r on u.user_id= r.user_id
),

```

```

get_users as
(
    select user_id from recommendation group by user_id having count(*)>1
),

grouped_podcast as
(
    select recommendation_type,next_recommendation_type, avg(next_click-
click_through_rate) as diff
    from lead_recommedation l join get_users g on l.user_id=g.user_id
    group by recommendation_type,next_recommendation_type

)
select * from grouped_podcast
order by diff desc ;

```

	recommendation_type	next_recommendation_type	diff
1	Video	Video	0.154285714285714
2	Blog	Video	0.12
3	Blog	Podcast	0.062
4	Blog	Blog	0.0258823529411765
5	Video	Podcast	0.0238095238095238
6	Podcast	Podcast	-0.00379310344827586
7	Podcast	Video	-0.0304166666666667
8	Podcast	Blog	-0.174375
9	Video	Blog	-0.220869565217391

#### Insights:

*Video → Video and Blog → Video transitions drive strong CTR, especially on Fridays and Tuesdays.*

*Video → Blog and Podcast → Blog consistently show negative CTR, disrupting engagement.*

*Same-type transitions (e.g., Video → Video) maintain high engagement across all days.*

*Fridays show strong performance across transitions, while Wednesdays favor Video-first flows.*

#### Recommendations:

*Focus on Video-first recommendations, avoid Video → Blog, and optimize high-engagement days like Fridays for better transitions.*

**Scope:**

*Validate timestamps by ensuring standardized time zones, addressing session overlaps (e.g., late-night sessions), and analyzing time gaps between transitions. Check for peak activity hours (morning vs. evening) and data completeness to ensure transitions are accurate and reflective of user behavior.*