Report:

This report I will try to explain according to the **Business Intelligence and Data Engineer** Lifecycle.

* As a Business Intelligence and Data Engineer working for the Organization, I collaborated with the business users and gathered the requirements.

Initially, the data I got from the organization are  
**user\_activity\_data:** The data consists of user\_activity on the platform and has columns like user\_id, session\_id, session\_length, message\_sent, feedback\_rating, resources\_clicked

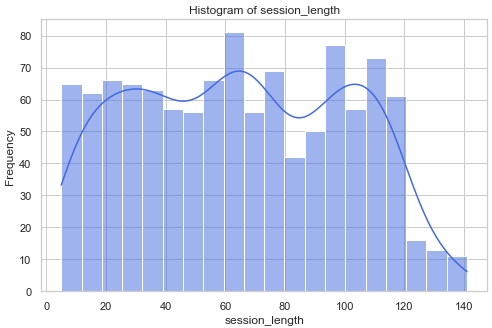
**recommendation\_data:** The data contain recommendations\_data and has columns like recommendation\_id, recommendation\_type, click\_through\_rate, feedback\_score

**moderator\_performance\_data:** The data contains moderators\_data and contains columns like moderator\_id, chat\_sessions\_moderated, avg\_response\_time, user\_satisfaction\_score

* As the next step, I would like to understand the data by plotting visualizations in matplotlib, seaborn using Python.

There are a lot of visualizations in the Jupyter notebook, to avoid congestion and keeping concise I’m listing only impactful visualizations.

**Distribution of session\_length:**



It's almost having the similar distribution for all of session length's. It means the users maintained between 10 to 120 minutes in the app.

**Distribution of messages\_Sent:**

A graph of a line graph

Description automatically generated with medium confidence

Most users sent between 10 to 40 messages per session.

**Distribution of feedback\_rating:**

**A graph and a pie chart

Description automatically generated**

Users given feedback between 1 to 6 where most of the users given 5(28.7%).

**Distribution of resources\_clicked:**

A close-up of a graph

Description automatically generated

It has almost the uniform distribution where users clicked between 1 to 4 resources.

**Distribution of recommendation\_type:**

A screenshot of a graph

Description automatically generated

Even the recommendations are uniform where Podcast's are more recommended for users compared to Video and blog.

**Coming to the User engagement analysis:**

A chart of different colored dots

Description automatically generated

This is the plot between messages\_Sent vs session\_length based on the feeback\_rating. Generally, the users who spent more time on the app should interact with the platform more and send more messages.

But we can clearly observe that, there are users who spent more time on platform passively and give very less feedback\_rating.

And there are users who spent less time on app and give more feedback. This feedback is lightly misunderstood. We need to deep dive more to create more features.

Distribution of average\_feeback\_rating and session\_length in buckets:

A graph of a number of different colored bars

Description automatically generated

This the average aggregated data across session\_length bins.

**Coming to the Recommendation analysis:**

I have analyzed the various recommendation\_type and its engagement  
A graph of a graph showing a number of blue and green bars

Description automatically generated with medium confidence

This picture clearly states that users are engaged with video type compared to blog and podcast.

On the other hand, the feeback\_Score for these recommendations are

A graph of a bar chart

Description automatically generated with medium confidence

It shows the uniform distribution of feedback\_Score among the recommendations with podcast is slightly less.

**Coming to the moderator performance analysis**

**A graph of blue bars

Description automatically generated with medium confidence**

Generally, it would be like if the moderator responds to the user in the very less time, he should be given more satisfaction\_score. Here it is complete viceversa.

It can be due to the moderator may not satisfied the user or he is closing the requests soon.

It is clearly stated that, if the moderator is responding late, he is getting less user\_Satisfaction\_Score.

**Joining the data:**

1. Joining all the tables is the difficult thing as we didn’t have the similar joining key in the tables.
2. Firstly, I have joined **users table** and the **recommendations table** based on the user\_id primary key.
3. Now, the challenge is joining the **user\_recommendation table** and the moderator table. To tackle this challenge, I have followed the proportional approach based on the idea where one moderator can assist different users.
4. So, we can generate moderators **by (chat\_moderators\_managed/total moderated\_sessions)\*100.**
5. Now we have list of moderators and we are joining them with the **users\_recommendation** table.

**Sample of final data**



Still there are few issues after the joins,

After joining the data, I found a lot of missing joins for recommendations table. Its 606 users, if there is less missing join rate, we can be able to provide better platform for user satisfaction.

A blue and red pie chart

Description automatically generated

Coming to the metrics,

To define the engagement or effectiveness across these datasets, I have defined three metrics. They are calculated per user and listed below.

1. **Engagement Score:** This metric is used to calculate engagement score by giving importance to session length and message length and giving importance to feedback rating. It is calculated as follows.

*Engagement score = (session\_length/max\_session\_length)\*0.4 + (messages\_sent/max\_messages\_sent)\*0.4 + (feedback\_rating/max\_feedback\_rating)\*0.2*

Here, the metric is used to evaluate user participation and satisfaction.

1. **Responsiveness Efficiency:** This metric is used to calclulate responsiveness efficieny by dividing avg\_reponse\_time and user\_satisfaction\_score. Finally diving by 5 to normalize it.

Responsiveness Efficiency = (1/(avg\_response\_time+1))x(user\_satisfaction\_Score/5)

1. **Recommendation Conversion Rate (RCR):** This metric is used to calculate the engaged recommendations per total users who received recommendations. The engaged recommendations are someone who has click\_through\_rate>0 and feedback\_score> median\_feedback\_Score to ensure that feedback\_score is not randomly given.

RCR = (engaged\_users/total\_recommendation\_users)x100

This metric meaningful user engagement with recommendations by focusing on both click-through rates and feedback quality.

I have created a separate python script to create the metrics based on the data. It should be run after the data cleaning script.

**Sample metrics image:**

A screenshot of a computer

Description automatically generated