Final Solution:
Conflict Analytics Lab

OpenJustice

Contents

OpenJustice

Project Overview

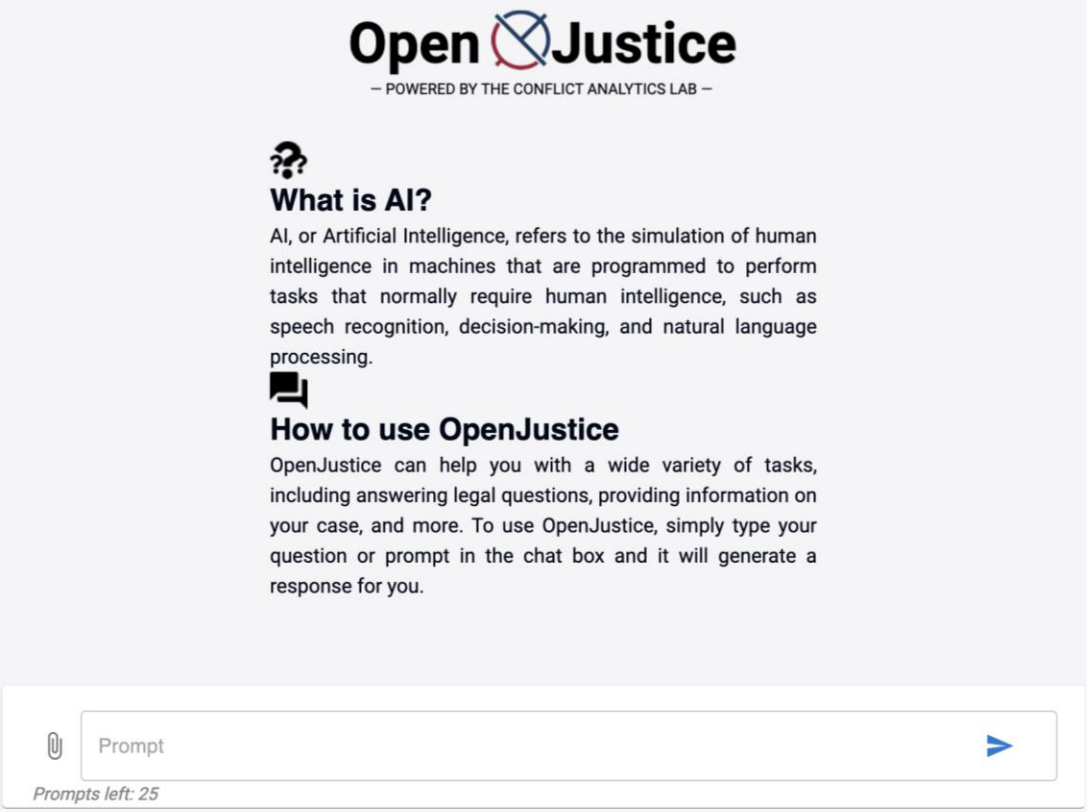
The OpenJustice AI, developed by the Conflict Analytics Lab, aims to transform the legal field by offering advanced AI-driven solutions to assist professionals in addressing legal issues more efficiently.

By enhancing the OpenJustice AI platform, the team aims to bridge the gap between legal expertise and technology, ensuring that users receive high-quality, actionable legal insights. The improvements will also contribute to the broader goal of making legal assistance more accessible and effective for users.

This community-focused project aims to assist the team in analyzing existing data and researching potential future implementations. The project concentrates on two main areas: an easy-to-follow dashboard which defines KPIs from current Google Analytics metrics and monitors the quality of AI-generated legal responses, and a user feedback mechanism proposal.

Note: link to OpenJustice AI https://openjustice.ai/

OpenJustice AI User Interface Page



OpenJustice

Value Proposition

Solution → Clients

Features

1. **Dashboard**: Provides simplified KPIs and metrics in a user-friendly interface.
2. **User Feedback Mechanism**: Based on UX/UI standards, ensuring continuous user experience improvements.
3. **PoC Code Demo**: Robust evaluation system using Vertex AI metrics for legal AI performance, ensuring high-quality responses.

Benefits

1. Simplified analytics and reporting through a consolidated dashboard.
2. Future improvements in user experience lead to higher engagement and satisfaction.
3. Help further fine-tuning of the model to ensure the quality of AI responses.
4. Clear and actionable insights driving informed decision-making and continuous improvement.

Needs

1. Concise and easily understandable metrics for assessing platform performance.
2. A robust evaluation system for AI performance.
3. Efficient user feedback collection and analysis.

Fears

1. Complex data reports leading to decision-making delays.
2. Quality and reliability of AI-generated legal responses.
3. Negative user experience driving users away.

Wants

1. A simplified, user-friendly dashboard with KPIs.
2. Enhanced UXUI for their platform.
3. High-quality, reliable, and well-formatted AI responses to legal questions.
4. Effective user feedback mechanisms to continually improve the platform.

Note: link to reference template https://www.peterjthomson.com/2013/11/value-proposition-canvas/

OpenJustice

KPI/Metric Development I

	Google Analytics	Importance
KPIs	1. **Acquisition KPIs:** ○ Number of new users	• Tracks the growth of the user base and the effectiveness of marketing strategies. • Helps understand user acquisition patterns and the success of outreach efforts.
	2. **Engagement KPIs:** ○ Bounce rate ○ Engagement rate ○ Number of engaged sessions	• Measures user interaction with the platform. • Identifies areas for improvement in user engagement. • Provides insights into user behavior and platform usability.
	3. **Conversion KPIs:** ○ Conversion rate	• Tracks the effectiveness of the platform in achieving desired outcomes. • Measures how well the platform converts visitors into users or other target actions.
	4. **Retention KPIs:** ○ Retention rate ○ Churn rate	• Monitors user loyalty. • Helps develop strategies to reduce churn and improve user retention.
	5. **UXUI KPIs:** ○ Net promoter score ○ Customer satisfaction score	• Guides strategic decisions to prioritize enhancements that ensure a positive and user-centric platform experience. • Foster user loyalty and satisfaction, and ultimately maximizing its societal benefits.

Note: 1) all the KPIs with underline are those being proposed and future implementation.
2) See Appendix A, B, and C for all definitions of KPIs and metrics.

OpenJustice

KPI/Metric Development II

	Google Analytics	Importance
Metrics	1. Total number of users 2. Total number of sessions 3. Average session duration 4. User stickiness 5. Total user engagement time 6. Traffic attribution 7. User segments (country, language, age, gender, etc.) 8. Overall GA Score (customized calculation) ○ 20% * #New users + 20% * #Engaged sessions + 30% * Bounce rate + 30% * Engagement rate 9. <u>Misunderstanding rate</u> 10. <u>Non-response rate</u> 11. <u>Number of follow-up questions in each conversation</u>	Included in dashboard (1-8): • Help to track the growth and scale of the user base. • Indicate user engagement levels and helps in understanding the utilization of the platform over time. • Indicate user loyalty, satisfaction, and engagement. • Provide a holistic view of platform performance by combining key engagement metrics into one easily understandable figure. It simplifies the assessment of overall platform health and user engagement. Proposed (9-11): • Indicate areas where the AI needs improvement, ensuring better quality responses in the future. • Address gaps in the AI's knowledge base, leading to a more reliable system. • Indicates the completeness and clarity of initial AI responses.

Note: all the metrics with underline are those being proposed and future implementation.

OpenJustice

KPI/Metric Development III

Metrics	Conversation Data	Importance
	1. Question answering (main focus): ○ question_answering_quality ○ QuestionAnsweringHelpfulness ○ QuestionAnsweringCorrectness ○ QuestionAnsweringRelevance	• Evaluate the model's ability to provide accurate and relevant legal answers. • Ensures the platform delivers high-quality information to users.
	2. General text generation: ○ Bleu ○ Rouge ○ Coherence ○ Fluency ○ Groundedness ○ Fulfillment	• Evaluate the model's ability to ensure the responses are useful, safe, and effective for users. • Evaluates the overall quality of text generation by the AI.
	3. Summarization: ○ summarization_quality ○ summarization_helpfulness ○ summarization_verbosity	• Potential future implementation for evaluating model summarization capabilities.

Note: all the metrics with underline are those being proposed and future implementation.

OpenJustice

User Feedback Mechanism Proposal I

Category	Question	Where to add	Format	Importance
General	How likely are you to recommend our OpenJustice AI to a friend or colleague?	Separate user feedbac section on web	11-point 0-6: detractors (unhappy customers) 7-8: passives (neutral customers) 9-10: promoters (happy customers)	• To calculate Net Promoter Score KPI (Calculation: subtract the percentage of detractors from the percentage of promoters.)
	How satisfied were you with your experience with our OpenJustice AI?	Separate user feedbac section on web	1-5 Likert Scale 1. Very Dissatisfied: The user is very unhappy with the product/service. 2. Dissatisfied: The user is unhappy with the product/service. 3. Neutral: The user feels indifferent about the product/service. 4. Satisfied: The user is happy with the product/service. 5. Very Satisfied: The user is very happy with the product/service.	• To calculate Customer Satisfaction Score KPI (Calculation: the sum of all positive responses, divided by the total responses collected, then multiplied by 100.)
	What is your overall impression of our OpenJustice AI?	Separate user feedbac section on web	Text	• Gathering qualitative data so we can perform text analysis to drive informed decision making.
	What changes or enhancements would most improve your overall satisfaction with our platform?	Separate user feedbac section on web	Text	• Encouraging detailed feedback to identify platform's strength and weakness.

Note: these will be implemented by the CA Lab team in near future.

OpenJustice

User Feedback Mechanism Proposal II

Category	Question	Where to add	Format	Importance
Content	The answers provided are clear in format and understandable.	After completing a conversation	1-5 Likert Scale 1. Strongly Disagree 2. Disagree 3. Neutral 4. Agree 5. Strongly Agree	• Evaluate the model performance by single conversation to get more accurate feedback and to know if a certain field of topics need to be fine-tuned.
	The responses generated are relevant and helpful to my queries.	After completing a conversation	Same as above	• Ensures the AI meets user expectations and fulfills its purpose.
	The content is up-to-date and accurate.	After completing a conversation	Same as above	
	Appendix D question form	Separate user feedback section on web	Text + multiple choice	• Detailed feedback from the user and future opportunities of doing user interviews to improve the model. • Helps to define what would be the most frequent issue for the AI model.

Note: these will be implemented by the CA Lab team in near future.

OpenJustice

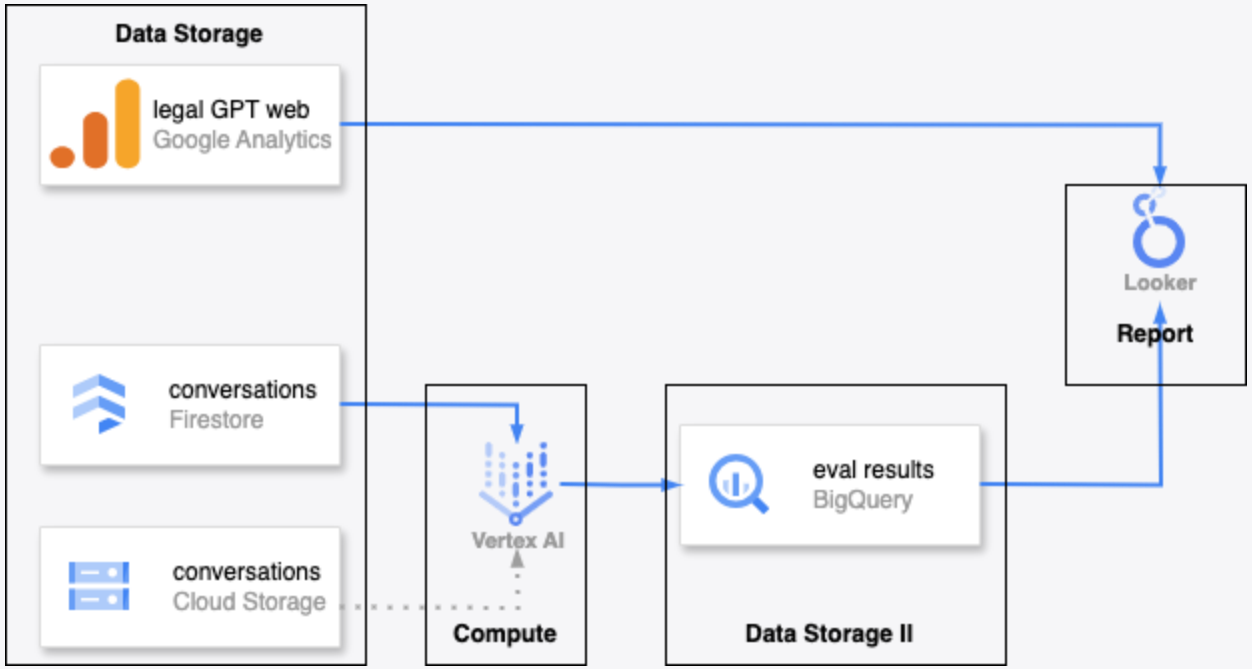
User Feedback Mechanism Proposal III

Category	Question	Where to add	Format	Importance
Webpage performance	The website is stable and loads quickly.	After completing a conversation	1-5 Likert Scale 1. Strongly Disagree 2. Disagree 3. Neutral 4. Agree 5. Strongly Agree	• Evaluate the webpage performance by single interaction to get more accurate feedback and to better maintain the website.
	The functions (Chat History, Upload PDF, etc.) are easy to follow.	After completing a conversation	Same as above	• Ensure users to access relevant information quickly and efficiently. • Enhances user satisfaction and reduces frustration
	Appendix E question form	Separate user feedback section on web	Text + multiple choice	• Detailed feedback from the user and future opportunities of doing user interviews to maintain the webpage. • Helps to define the technical issues encountered while using the website.

Note: these will be implemented by the CA Lab team in near future.

OpenJustice

Technology Architecture



Note: Google Cloud platform is used to align with their current working environment.

Data Flow:

1. Data Collection and Storage: Google Analytics (raw), Firestore (raw), and BigQuery (processed)
2. Data Processing: Vertex AI (conversation data only)
3. Data Reporting: Looker Studio

Architecture Layers:

- Storage:
  - **Google Analytics:** tracking and analyzing user behavior on the platform.
  - **Firestore:** the primary database for storing legal AI conversational data.
  - **BigQuery:** serving as a data warehouse to store the evaluation results from code notebook.
  - **Cloud Storage (optional - this was setting up in a personal account for testing purposes using Excel files):** storing datasets and files.
- Compute (Appendix F, G, H, and I for code breakdown):
  - **Vertex AI:** powers evaluating and improvingimprovement of the legal AI's performance.
- Report:
  - **Looker Studio:** data visualization and reporting through interactive dashboards, which containing both Google Analytics metrics and conversational data evaluation results.

OpenJustice

UXUI – End Users

	Pain Points	Solution
Developers	• Difficulty in interpreting historical data. • Lack of real-time feedback on model performance. • Inadequate tools for measuring and improving AI performance.	• Provide customized aggregated metrics for easier interpretation of the Google Analytics data. • Offer real-time dashboards of conversational data quality. • Integrate with existing development tools and platforms.
Project Managers	• Overload of reporting data that is difficult to interpret or redundant. • Challenges in getting feedback about the current website performance to improve.	• Create high-level dashboards with key KPIs and metrics. • Provide clear, simplified proposals for future implementation or direction of the project.
Other Stakeholders	• Summarized insights that are easy to understand. • Clear visual representations of project impact and performance.	• Offer easy-to-understand summaries and visualizations. • Ensure accessible and user-friendly interfaces.

Note: the Developers and Project Manager are mentioned here because they were the main point of contact during this community project.

OpenJustice

UXUI – Dashboard Overview

1. Home Page
   - Introduction of the dashboard pages and their usage.
   - Current user portfolios to help understand user segments.
2. Google Analytics Overview
   - Main KPIs and metrics defined earlier with comparisons with past periods.
3. Google Analytics Aggregated KPI
   - Breakdown of the customized KPI, Overall GA Score (which KPIs are included to calculated this value and their trends).
4. Conversation Evaluation Metrics (BigQuery)
   - The evaluation results of the conversational data (3 aspects).

Note: the direction arrows indicate the connections between pages when clicking on the text.

Note: 1) link to dashboard views https://lookerstudio.google.com/embed/reporting/46996a46-3e99-49b5-b038-b1b5b1837740/page/p_eatbulqajd

2) Appendix J to U for user guide and more explanations

OpenJustice

Sustainability – Current State

Documentations

- KPI and metrics development are proposed or added based on the real-time data from Google Analytics or standard essential metrics for similar products (e.g. webpage, GPT models).
- The user feedback mechanism is proposed for future implementation, which enhances the functionality of the web and future user experience.
- A detailed user manual for the dashboard that includes explanations of the elements and instructions on how to modify them is included, and it is easy to follow. Placeholders for proposed features are included to scalability.
- This summary document, along with other detailed supporting documents are shared and organized for clients' references.
- A final handover meeting will be held to walk through the entire solution and address any last-minute questions or concerns.

Technology

All these tools are being implemented or already used based on the consideration of the client's current working environment with the Google platform, which ensures the sustainability of easier updates and maintainance.

- **Google Analytics:** comprehensive data on website performance and user behavior.
  - Easily maintainable with existing support resources and extensive documentation.
- **Firestore:** scalable NoSQL database for storing and syncing data in real-time.
  - Fully managed by Google, ensuring high availability and automatic scaling. Minimal maintenance required from the client's side.
- **Vertex AI: a**dvanced machine learning and AI capabilities.
  - Regularly updated by Google to include the latest AI advancements. Extensive support and documentation available for continued use.
- **BigQuery:** data warehousing solution for querying large datasets and is easy to connect with Looker Studio.
  - Fully managed, with automatic scaling and maintenance. Cost-effective with pay-per-query pricing.
- **Looker Studio: a v**isualization tool for creating interactive and shareable dashboards.
  - User-friendly with extensive support and community resources. Free to use without extra charge to maintain a cost efficiency.

OpenJustice

Sustainability – Future State

Enhanced Features

- Implement the proposed KPIs and metrics as the project evolves and more data becomes available.
- Introduce new analytical models and visualization tools to provide deeper insights.

AI Model Updates

- Continuously update and fine-tune the AI models based on the evaluation results on historical data and newly implemented user feedback.
- Integrate more advanced machine learning techniques to enhance the quality of responses.

User Experience Enhancements

- Regularly update the UX/UI based on user feedback and usability studies.
- Expand the user feedback mechanism to capture more detailed insights (use the proposed mechanism as a starting point).
- A/B testing can be implemented for research purposes to define the effectiveness of the proposals from this project.

OpenJustice

Appendix

- KPI/Metrics Definitions
- User Feedback Detailed Forms
- Code Breakdown
- Dashboard Guide

OpenJustice

A - Google Analytics KPIs' Definitions

1. **Acquisition KPIs**:
   - Number of new users
2. **Engagement KPIs:**
   - Bounce rate: the percentage of users who leave the chatting page without interacting.
   - Engagement rate: engaged sessions / total sessions.
   - Number of engaged sessions: the number of sessions that lasted 10 seconds or longer, or had 1 or more key events or 2 or more page or screen views.
3. **Conversion KPIs:**
   - Conversion rate: the percentage of users who complete a desired action (e.g., sign up, make an engagement).
4. **Retention KPIs:**
   - Retention rate: the percentage of users who continue using the platform over a given timeframe.
   - Churn rate: the rate at which users stop using the platform over a specific period.
5. **UXUI KPIs:**
   - Net promoter score: how satisfied users were with their overall experience.
   - Customer satisfaction score: how well a product or service fulfills users' expectations

Note: 1) all the metrics with underline are those being proposed and future implementation.
2) link to the Google Analytics reference website
https://support.google.com/analytics/answer/13947485?hl=en&sjid=16883852291625120365-NA&visit_id=638572226648757371-3016463232&ref_topic=13948007&rd=1

OpenJustice

B - Google Analytics Metrics' Definitions

1. Total number of users
2. Total number of sessions
3. Average session duration
4. User stickiness: DAU/WAU shows the percentage of users who engaged in the last 24 hours out of the users who engaged in the last 7 days.
5. Total user engagement time: the amount of time users spend with the web page in focus
6. Traffic attribution: channel distribution for session traffic including the number of active users, engagement rate and time with comparisons with the past period
7. User segments (country, language, age, gender, etc.)
8. Overall GA Score: a customized aggregated value by giving certain KPIs different weights to measure the performance.
    - 20% * #New users + 20% * #Engaged sessions + 30% * Bounce rate + 30% * Engagement rate
9. <u>Misunderstanding rate:</u> estimates how frequently the AI misunderstands user queries, leading to incorrect responses.
10. <u>Non-response rate: t</u>he number of times the AI has failed to push some content following a user question (due to lack of content or misunderstanding).
11. <u>Number of follow-up questions in each conversation</u>

Note: all the metrics with underline are those being proposed and future implementation.

OpenJustice

C - Conversation Data Metrics' Definitions

1. **Question answering (main focus)**:
   - question_answering_quality: describes the model's ability to answer questions given a body of text to reference.
   - QuestionAnsweringHelpfulness: describes the model's ability to provide important details when answering a question.
   - QuestionAnsweringCorrectness: describes the model's ability to correctly answer a question.
   - QuestionAnsweringRelevance: describes the model's ability to respond with relevant information when asked a question.
2. **General text generation**:
   - Bleu: holds the result of an algorithm for evaluating the quality of the prediction.
   - Rouge: compares the provided prediction parameter against a reference parameter
   - Coherence: describes the model's ability to provide a coherent response.
   - Fluency: describes the model's language mastery.
   - Groundedness: describes the model's ability to provide or reference information included only in the input text.
   - Fulfillment: describes the model's ability to fulfill instructions.
3. **Summarization**:
   - summarization_quality: describes the model's ability to summarize text.
   - summarization_helpfulness: describes the model's ability to satisfy a user's query by summarizing the relevant details in the original text without significant loss in important information.
   - summarization_verbosity: measures if a summary is too long or too short.

Note: 1) all the metrics with underline are those being proposed and future implementation.
2) link to the Google Vertex AI reference website https://cloud.google.com/vertex-ai/generative-ai/docs/models/determine-eval

OpenJustice

D - Content Feedback Detailed Form

1. Email
2. System message and chat log
3. What were you expecting from the completion?
4. Why is the model output not ideal?
    - The model isn't adhering to the system message
    - The model's response is inaccurate
    - The model's response is not useful
    - Other
5. Please provide more details of why the output is not ideal. For instance, what is inaccurate about the response?
6. Would you be interested in being contacted for further collaboration with our research team?
    - Yes
    - No
7. Is there anything else you'd like to share about your experience?

Note: link to reference form from ChatGPT https://openai.com/form/chat-model-feedback/

OpenJustice

E - Webpage Feedback Detailed Form

1. Email
2. System error message
3. What was your last action/navigation?
4. Why is the webpage not performing well?
    - The loading time is too long
    - The responding time is too long
    - The webpage does not generate any response
    - The webpage is not saving the conversation as a chat history
    - <u>Other future function issues</u>
    - Others
5. Please provide more details of the problems that you have encountered. For instance, if a certain legal question is not getting any responses.
6. Would you be interested in being contacted for further collaboration with our research team?
    - Yes
    - No
7. Is there anything else you'd like to share about your experience?

Note: the option with underline is being proposed and future implementation.

OpenJustice

F - Initialization

```python
# Libraries
import logging
from google.cloud import aiplatform
from IPython.display import display, HTML, Markdown
import plotly.graph_objects as go
import nest_asyncio
import warnings
import vertexai
from vertexai.preview.evaluation import (
    EvalTask,
    CustomMetric,
    make_metric,
)
import langid
from google.cloud import storage
from google.cloud import bigquery
import pandas as pd
import io
import numpy as np
from ast import literal_eval
from google.auth import default

import firebase_admin
from firebase_admin import credentials, storage, firestore
```

```python
# Initialize Vertex AI and BigQuery
vertexai.init(project='[redacted]', location='[redacted]')
client = bigquery.Client()

# Setup logging and warnings
logging.getLogger("urllib3.connectionpool").setLevel(logging.ERROR)
nest_asyncio.apply()
warnings.filterwarnings("ignore")
```

```python
# Initialize Firebase Admin SDK
cred = credentials.ApplicationDefault()
firebase_admin.initialize_app(cred)

# Initialize Firestore
firestore_client = firestore.client()
```

Note: the notebook is set up in client's Google Cloud with all the prerequisites installed.

1. SDK installation:
   - Google Cloud SDK: Ensure have the Google Cloud SDK is installed and configured on machine.
   - Python Packages: Install the required Python packages using pip.
2. Service Initialization and Configuration:
   - Google Cloud Platform (GCP) Setup:
     - Vertex AI: Initialize Vertex AI with project ID and location.
     - BigQuery: Ensure GCP project has BigQuery enabled and have access to it.
   - Firebase Setup:
     - Firebase Admin SDK: Initialize Firebase Admin SDK using application default credentials.
     - Firestore: Ensure Firestore is enabled in Firebase project.
     - Storage: Ensure Firebase Storage is enabled in Firebase project.

OpenJustice

G - Data Preprocessing

```python
# Function to fetch data from Firestore and convert it to DataFrame with "id" and "conversation" columns
def fetch_data_from_firestore():
    collection_ref = firestore_client.collection('conversations')
    docs = collection_ref.stream()

    # Collect documents into a list with "id" and "conversation" fields
    data = []
    for doc in docs:
        doc_dict = doc.to_dict()
        if 'conversation' in doc_dict:
            data.append({
                'id': doc.id,
                'conversation': doc_dict['conversation']
            })

    df = pd.DataFrame(data)
    return df

# Fetch data and create DataFrame
df = fetch_data_from_firestore()
df.head()
```

```python
# Language detection
def detect_language(text):
    return langid.classify(text)[0] == 'en'  # Returns True if language is English, False otherwise
df['is_english'] = df['question'].apply(detect_language)

# Keep rows where is_english is True
df_en = df[df['is_english'] == True]
df_other = df[df['is_english'] == False]

df_en.drop(columns=['is_english'], inplace=True)
df_other.drop(columns=['is_english'], inplace=True)
df_en
```

```python
# Preprocess the data
def safe_convert(value):
    return value if isinstance(value, list) else []

df["conversation"] = df["conversation"].apply(safe_convert)

def reformat_conversation(conversation):
    formatted_conversation = [
        {"question": msg["content"]} if msg.get("role") == "user" else {"answer": msg["content"]}
        for msg in conversation
    ]
    return formatted_conversation

df["conversation"] = df["conversation"].apply(reformat_conversation)

def extract_values(conversation_list, key):
    try:
        values = [msg[key] for msg in conversation_list if key in msg]
        return values[0] if values else None
    except Exception as e:
        print(f"Error extracting values for key '{key}': {e}")
        return None

df['question'] = df['conversation'].apply(lambda x: extract_values(x, 'question'))
df['answer'] = df['conversation'].apply(lambda x: extract_values(x, 'answer'))

# Drop rows where 'question' or 'answer' is NaN
df = df.dropna(subset=['question', 'answer'])
df.reset_index(drop=True, inplace=True)
df
```

1. Fetch the data from the Firestore "conversation" collection.
2. Reformat the "Conversation" column to a standard "question-answer" pair for future evaluation.
3. Seperate the "Conversation" column to two columns, "question" and "answer" each.
4. Since Vertex AI doesn't provide high accuracy with non English text, store the English conversation in a separate data frame and only evaluate those only.

OpenJustice

H - Evaluation

```python
# Helper functions
def display_eval_report(eval_result, metrics=None):
    """Display the evaluation results."""

    title, summary_metrics, report_df = eval_result
    metrics_df = pd.DataFrame.from_dict(summary_metrics, orient="index").T
    if metrics:
        metrics_df = metrics_df.filter(
            [
                metric
                for metric in metrics_df.columns
                if any(selected_metric in metric for selected_metric in metrics)
            ]
        )
        report_df = report_df.filter(
            [
                metric
                for metric in report_df.columns
                if any(selected_metric in metric for selected_metric in metrics)
            ]
        )

    # Display the title with Markdown for emphasis
    display(Markdown(f"## {title}"))

    # Display the metrics DataFrame
    display(Markdown("### Summary Metrics"))
    display(metrics_df)

    # Display the detailed report DataFrame
    display(Markdown(f"### Report Metrics"))
    display(report_df)
def display_explanations(df, metrics=None, n=1):
    style = "white-space: pre-wrap; width: 800px; overflow-x: auto;"
    df = df.sample(n=n)
    if metrics:
        df = df.filter(
            ["instruction", "context", "reference", "completed_prompt", "response"]
            + [
                metric
                for metric in df.columns
                if any(selected_metric in metric for selected_metric in metrics)
            ]
        )

    for index, row in df.iterrows():
        for col in df.columns:
            display(HTML(f"<h2>{col}:</h2> <div style='{style}'>{row[col]}</div>"))
        display(HTML("<hr>"))
```

```python
# Prepare evaluation dataset
eval_dataset = pd.DataFrame(
    {
        "instruction": df_en["question"],
        "response": df_en["answer"],
    }
)

# Evaluate
eval_task = EvalTask(
  dataset=eval_dataset,
  metrics=[
    #'question_answering_quality',
    'question_answering_relevance',
    'question_answering_helpfulness',
    #"summarization_quality",
    #"summarization_verbosity",
    #"summarization_helpfulness",
    'fulfillment',
    #'groundedness',
  ],
  experiment='[redacted]'
)

eval_result = eval_task.evaluate()
```

•••

```python
# Display evaluation results
display_eval_report((("Eval Result", eval_result.summary_metrics, eval_result.metrics_table)))
```

1. Two helper functions to display the evaluation report and example explanation.
2. Self-identified metrics (for now, only display those that can be done with only user questions and AI responses).

Note: link to the reference notebook from Google Vertex AI website https://colab.research.google.com/github/GoogleCloudPlatform/generative-ai/blob/main/gemini/evaluation/evaluate_rag_rapid_evaluation_sdk.ipynb#scrollTo=tdMpJnLKXvsk

OpenJustice

I - BigQuery

```python
# Initialize BigQuery client
bq_client = bigquery.Client()

# Define the schema
schema = [
    bigquery.SchemaField("question_answering_relevance_mean", "FLOAT"),
    bigquery.SchemaField("question_answering_relevance_std", "FLOAT"),
    bigquery.SchemaField("question_answering_helpfulness_mean", "FLOAT"),
    bigquery.SchemaField("question_answering_helpfulness_std", "FLOAT"),
    bigquery.SchemaField("fulfillment_mean", "FLOAT"),
    bigquery.SchemaField("fulfillment_std", "FLOAT"),
    bigquery.SchemaField("timestamp", "TIMESTAMP"),
]

# BigQuery table details
table_id = '[redacted]

# Create a DataFrame with the evaluation summary metrics
summary_metrics_df = pd.DataFrame([eval_result.summary_metrics])
summary_metrics_df['timestamp'] = pd.Timestamp.now()

# Round the summary metrics to two decimal places
summary_metrics_df = summary_metrics_df.round(2)

# Sanitize column names
summary_metrics_df.columns = summary_metrics_df.columns.str.replace('/', '_')

# Load summary metrics to BigQuery
job = bq_client.load_table_from_dataframe(summary_metrics_df, table_id)
job.result()
print(f"Data loaded to {table_id}.")
```

1. More metrics name can be added into "schema" later when they are evaluated.
2. Only load the summary metrics of the data file into BigQuery instead of each rows' result (so the dashboard will only display the average scores for the entire data).

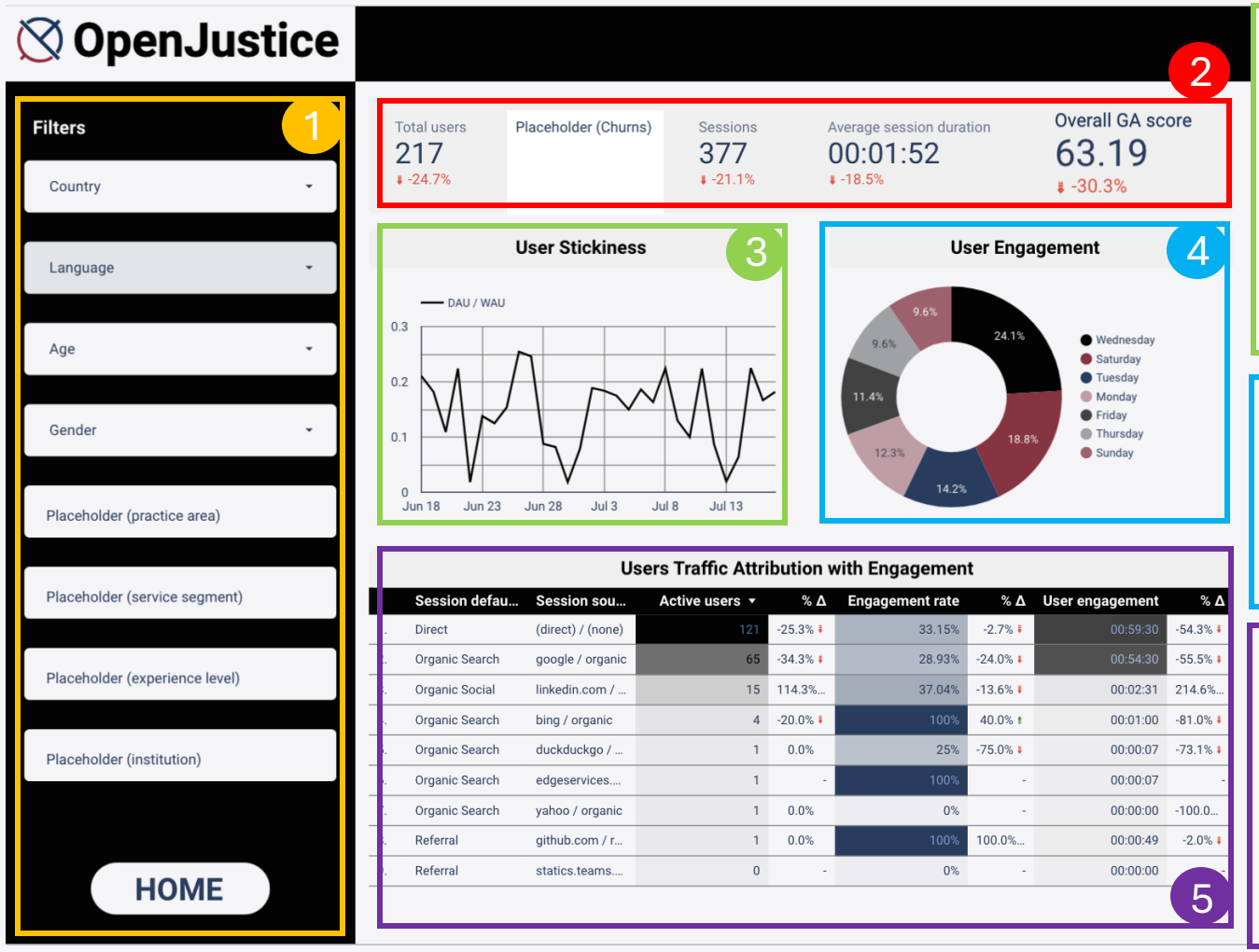
OpenJustice

J - Page 1 Overview

Scorecards:
Important metrics that contain data for the past 30 days with a comparison

Filters:
Including all types of segments.

1. Country and Language are currently available
2. Age and Gender don't have data stored
3. Other place holders are proposed to be included in the future



Session defau...	Session sou...	Active users	% Δ	Engagement rate	% Δ	User engagement	% Δ
Direct	(direct) / (none)	121	-25.3%	33.15%	-2.7%	00:59:30	-54.3%
Organic Search	google / organic	65	-34.3%	28.93%	-24.0%	00:54:30	-55.5%
Organic Social	linkedin.com / ...	15	114.3%...	37.04%	-13.6%	00:02:31	214.6%...
Organic Search	bing / organic	4	-20.0%	100%	40.0%	00:01:00	-81.0%
Organic Search	duckduckgo / ...	1	0.0%	25%	-75.0%	00:00:07	-73.1%
Organic Search	edgeservices....	1	-	100%	-	00:00:07	-
Organic Search	yahoo / organic	1	0.0%	0%	-	00:00:00	-100.0...
Referral	github.com / r...	1	0.0%	100%	100.0%...	00:00:49	-2.0%
Referral	statics.teams....	0	-	0%	-	00:00:00	-

User stickiness over time:
DAU/WAU shows the percentage of users who engaged in the last 24 hours out of the users who engaged in the last 7 days. A higher ratio suggests good engagement and user retention

User engagement by days:
The amount of time users spend with the web page in focus in each day of the week to see the distributions

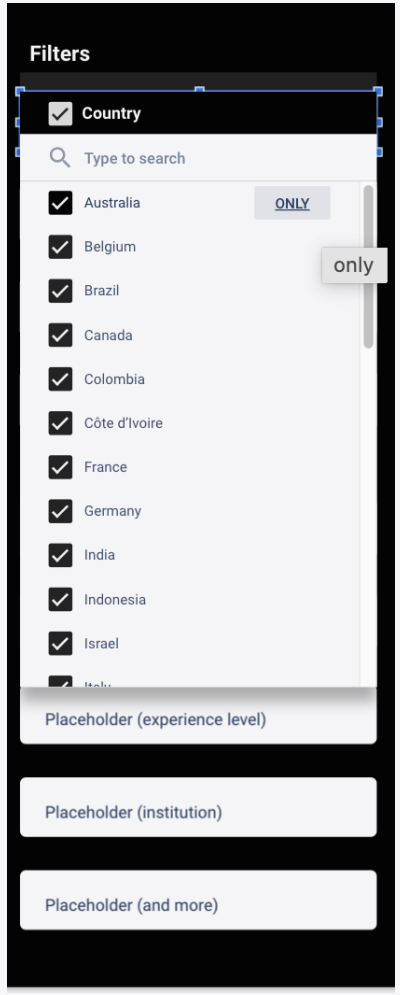
Traffic attribution for channels:
The breakdown of channel distribution for session traffic including the number of active users, engagement rate and time with comparisons with the past period

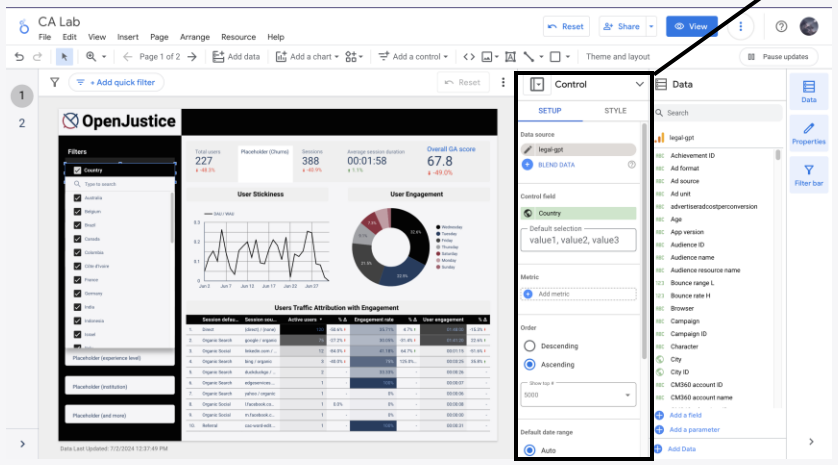
OpenJustice

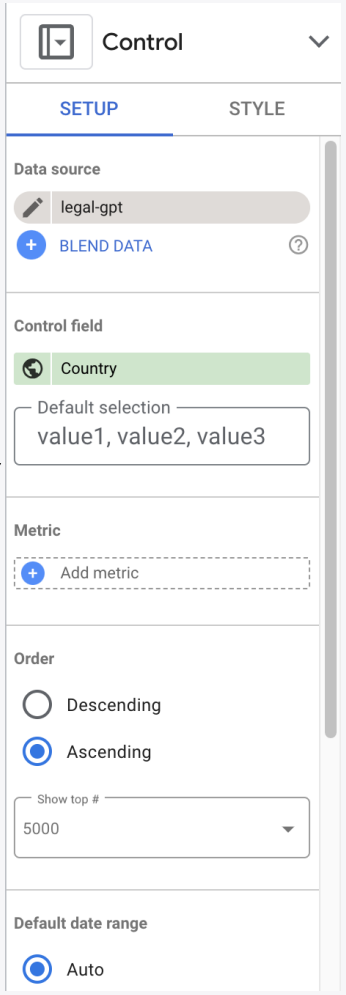
K - Page 1 Filters

To use:

1. Click on a certain filter to select the ones you want
2. Or to select the filter again to clear selections
3. Change the Placeholders by "Add a control -> Drop-down list" and change the "Control field", no Metric needed







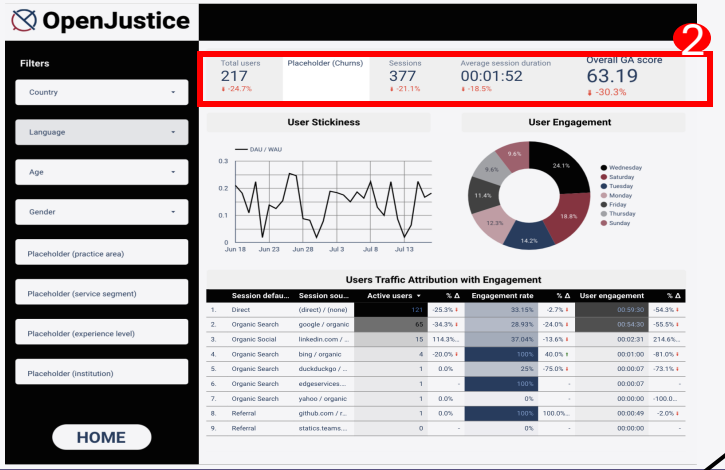
To edit: There will be a Control bar on the right that allows to make changes to the elements

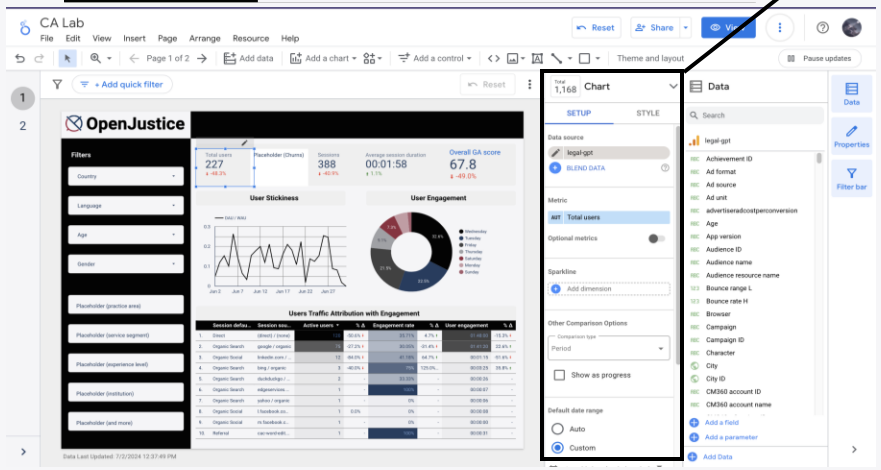
OpenJustice

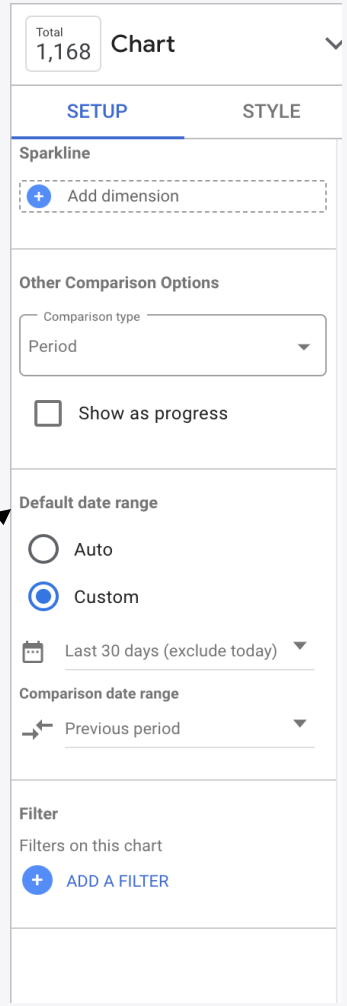
L - Page 1 Scorecards

1. Total users
2. Churn: number of users churning, and it can be used to calculate churn rate in the future (change the placeholder with a Scorecard by "Add a chart -> (1st) Scorecard" in the future)
3. Sessions: user interactions with the website that take place within a given time frame
4. Average session duration
5. Overall GA score: customized aggregation of several KPIs. Clicking on it will lead to the second page

** All the values are now set to the last 30 days, and they can be changed.



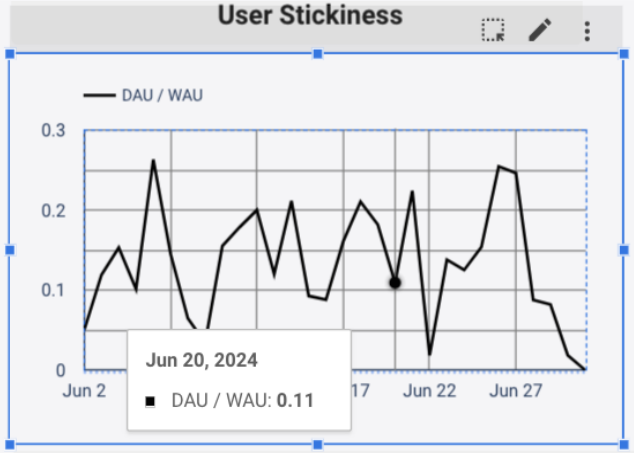


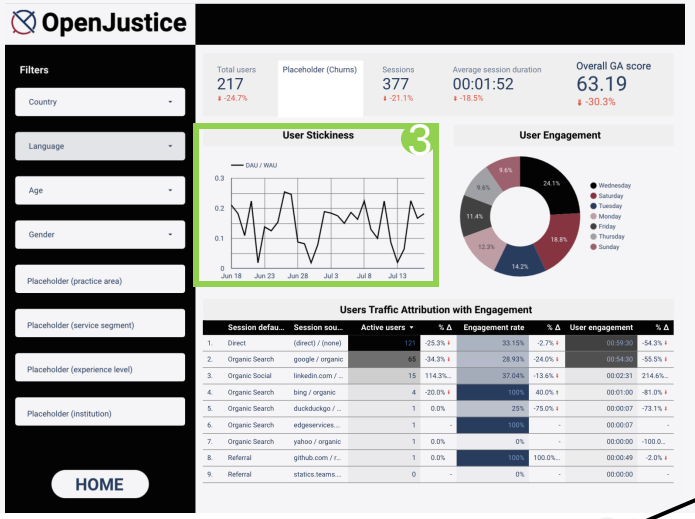


To edit:
Total user card as an example to change the time range, on the right Control bar, change "Default date range"

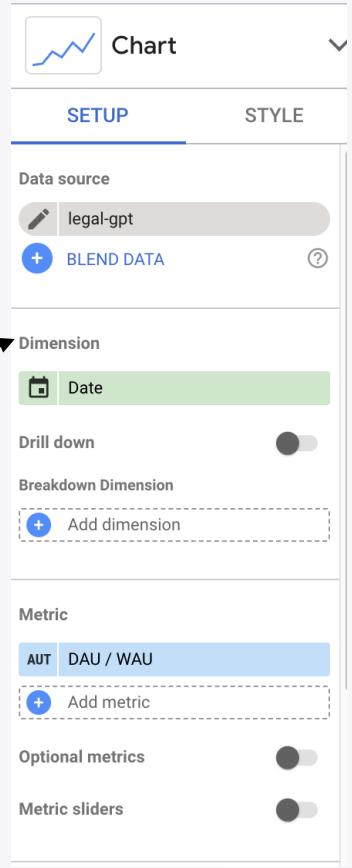
OpenJustice

M - Page 1 Line Chart





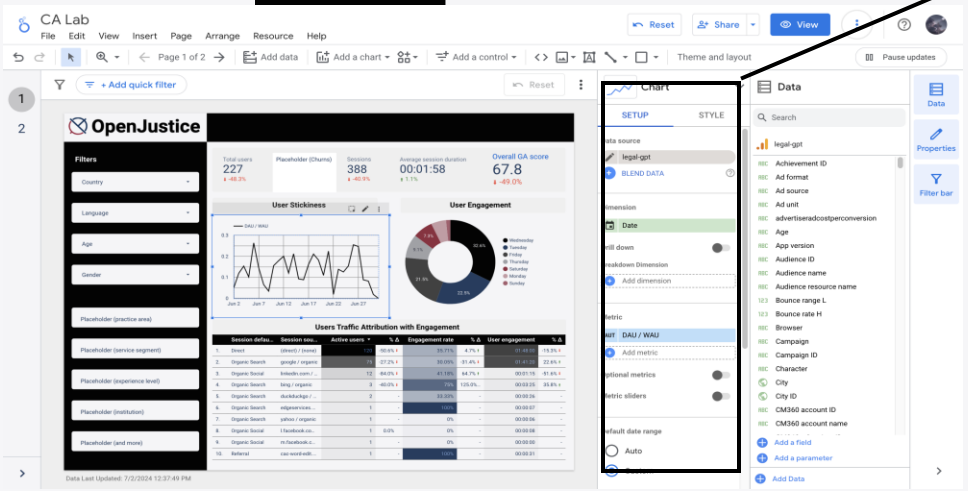
** The time range is now set to the last 30 days, and they can be changed.



To edit:

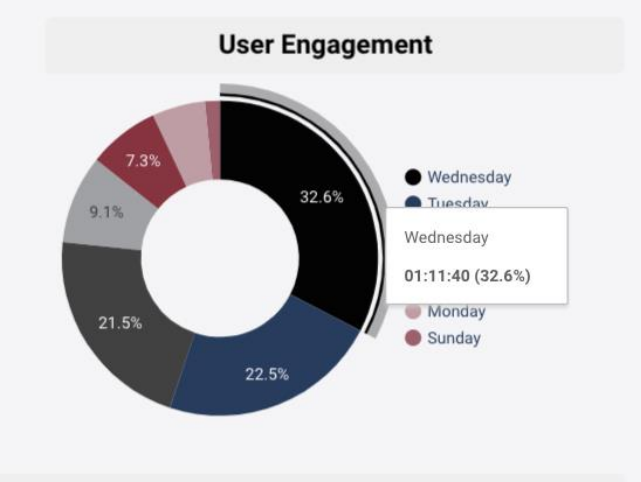
1. The same way to adjust time range on the right Control bar, change "Default date range"
2. Change "Dimension" to choose display date, month, and so on

To use:
Move the mouse to a certain point will show the value of a certain date

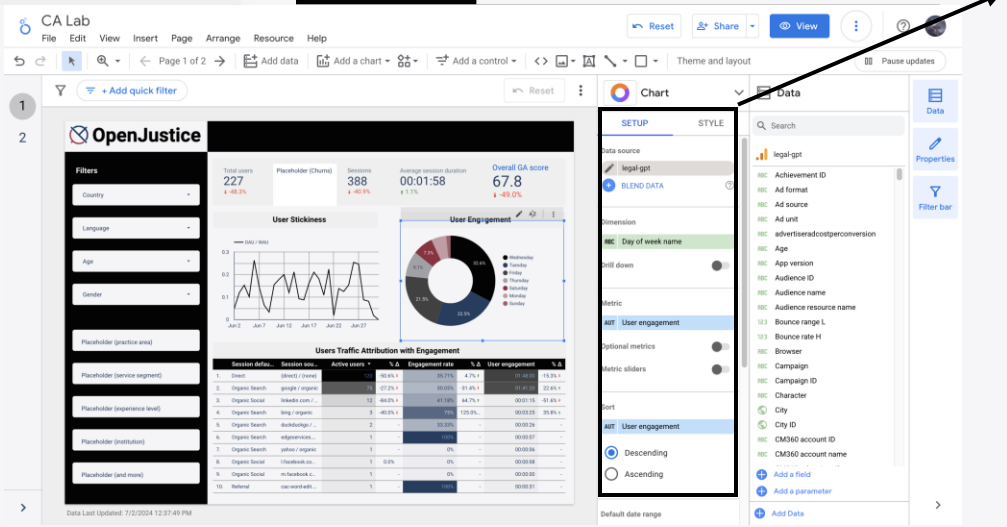


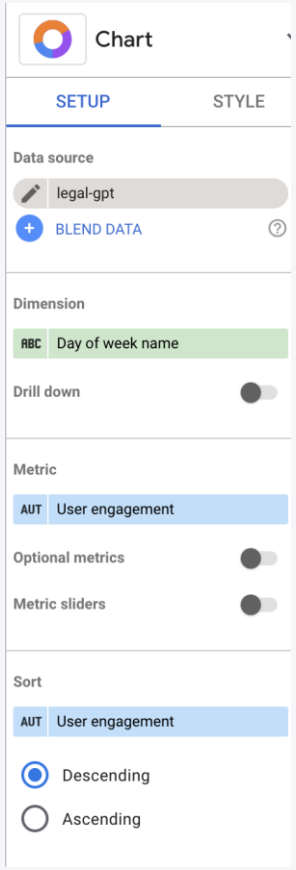
OpenJustice

N - Page 1 Pie Chart



** The time range is now set to the last 30 days, and they can be changed.





To edit:

1. The same way to adjust time range on the right Control bar, change "Default date range"
2. Change "Dimension" to choose to display the engagement time by different categories

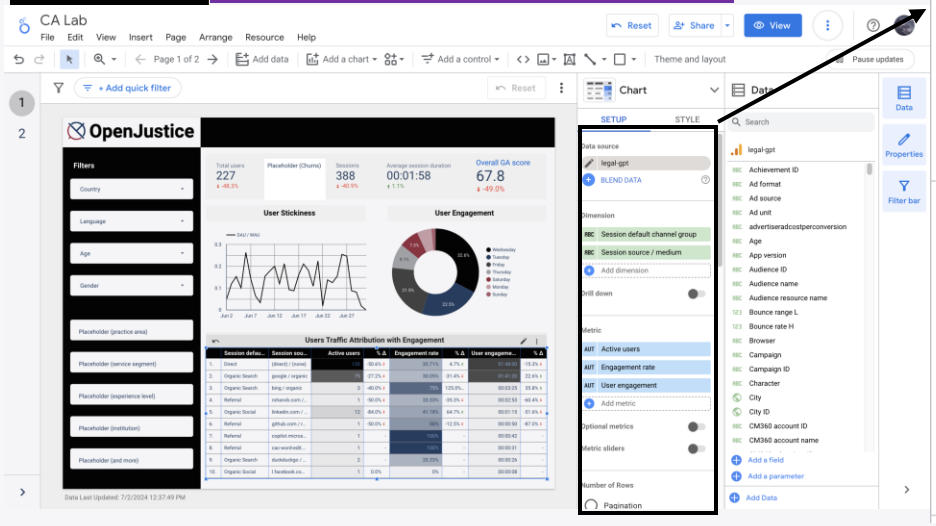
To use:
Move the mouse to a certain part will show the value of a certain day of the week

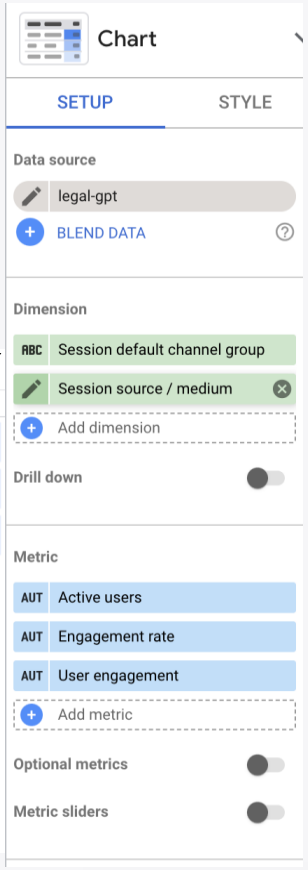
OpenJustice

O - Page 1 Table

1. Session default channel group: the channels by which users arrived at the site when they initiated new sessions
2. Sessions source and medium: source is where the website's traffic comes from (individual websites, Google, Facebook etc). Medium is how it got there (organic traffic, paid traffic, referral etc), which can help understand where our users came from
3. Active users
4. Engagement rate: the percentage of engaged sessions
5. User engagement

** The time range is now set to the last 30 days, and they can be changed.





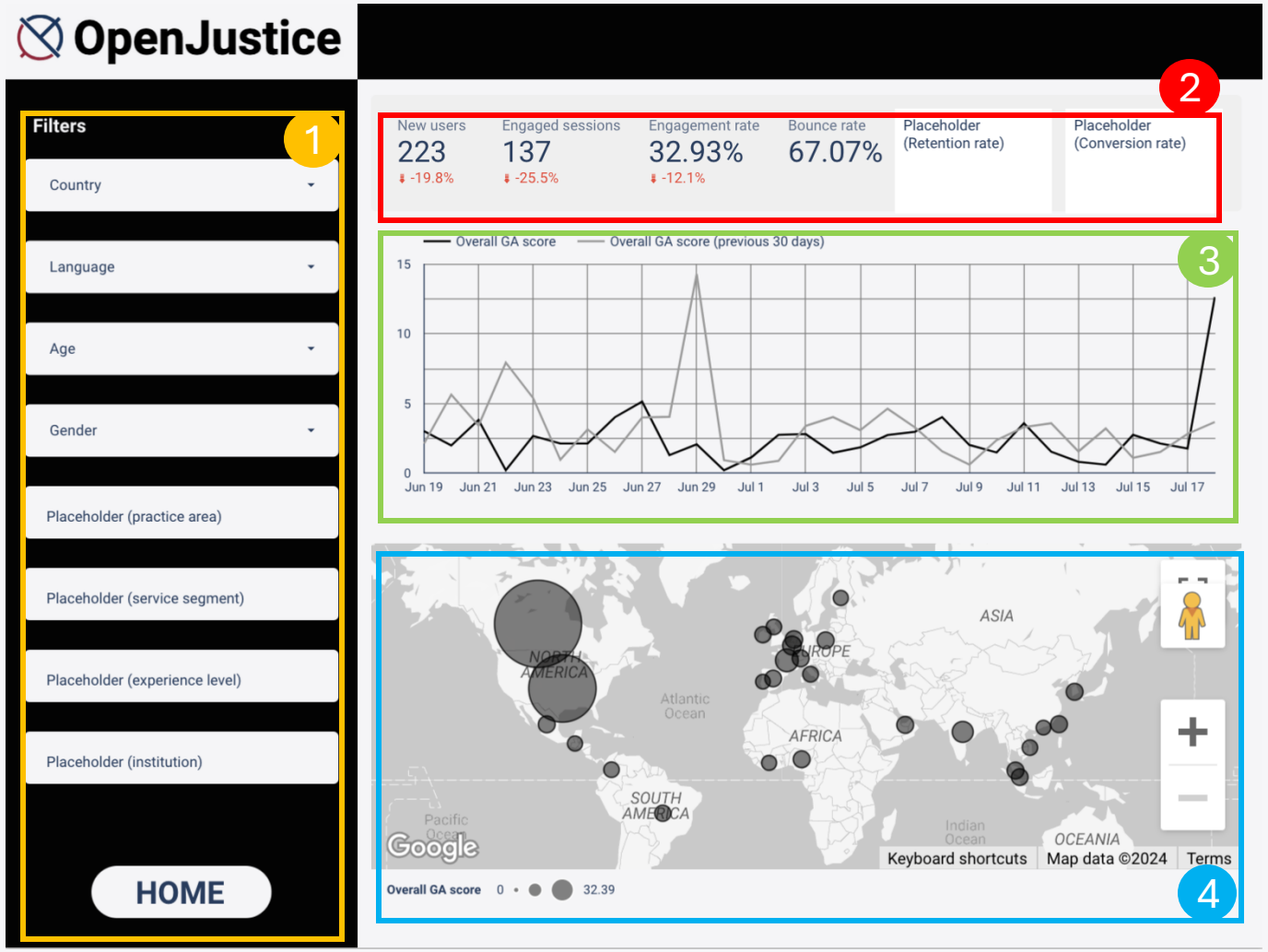
To edit:

1. The same way to adjust time range on the right Control bar, change "Default date range"
2. Change what dimensions and metric to include

OpenJustice

P - Page 2 Overview

Filters: The same filter as page 1



Scorecards:
Important KPIs used to calculate the Overall GA score, each contains data for the past 30 days with a comparison of the last period

Overall GA score over time:
Now, it uses 4 KPIs and weights of New users (20%), Engaged sessions (20%), Engagement rate (30%), and Bounce rate (30%) to calculate the score. The KPIs used and weights assigned can be changed

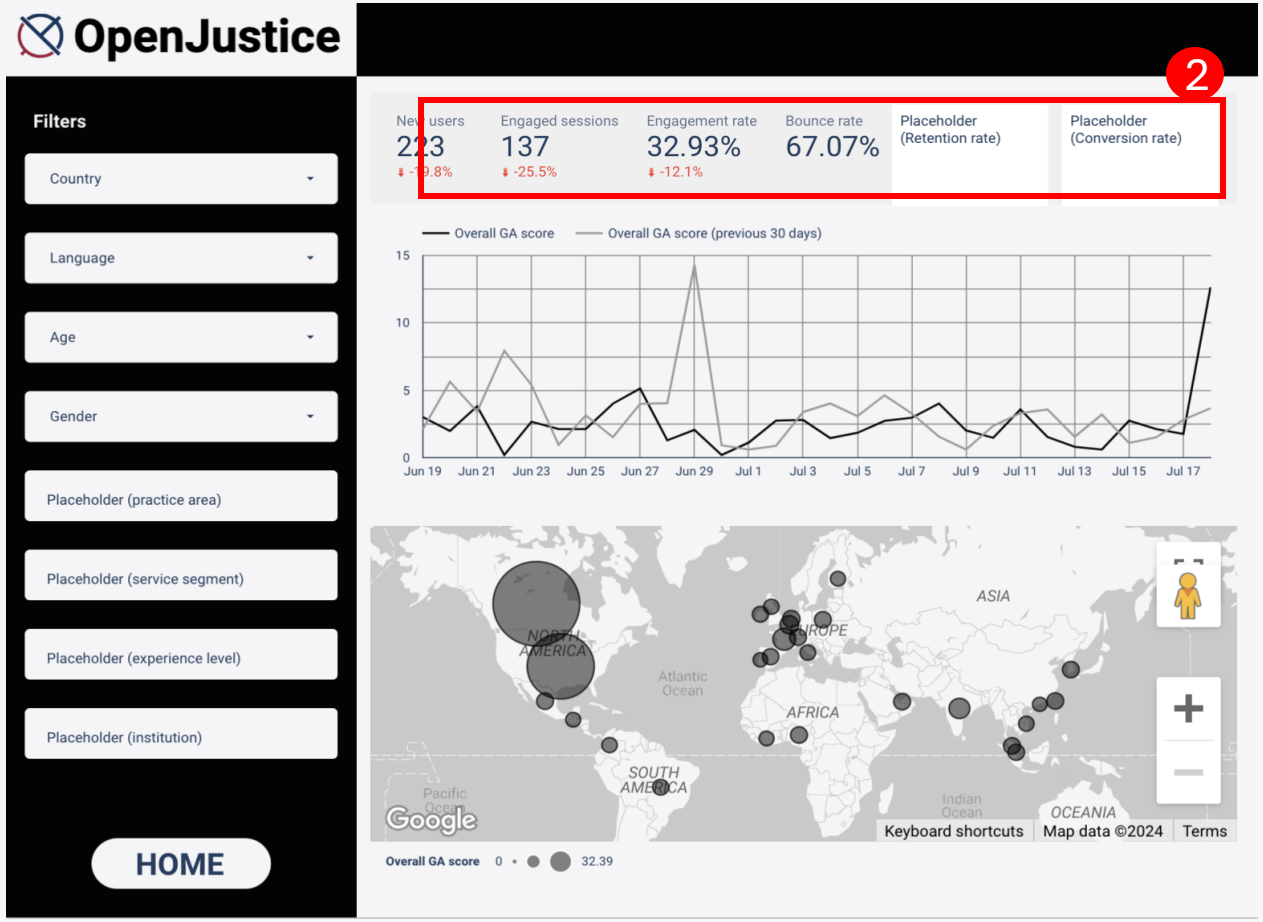
User maps:
The visualization of the user distribution geographically, and larger dot represents a higher density. The map will zoom in if a certain country is selected from the filter on the left

OpenJustice

Q - Page 2 Scorecards

1. New users
2. Engaged sessions: a session that lasts longer than 10 seconds, has a key event**, or has at least 2 pageviews or screen views
3. Engagement rate
4. Bounce rate: the percentage of sessions that were not engaged
5. Retention rate: shows the percentage of users who return each day in their first 42 days
6. Conversion rate: the percentage of users who complete a desired action (e.g., sign up, make an engagement)

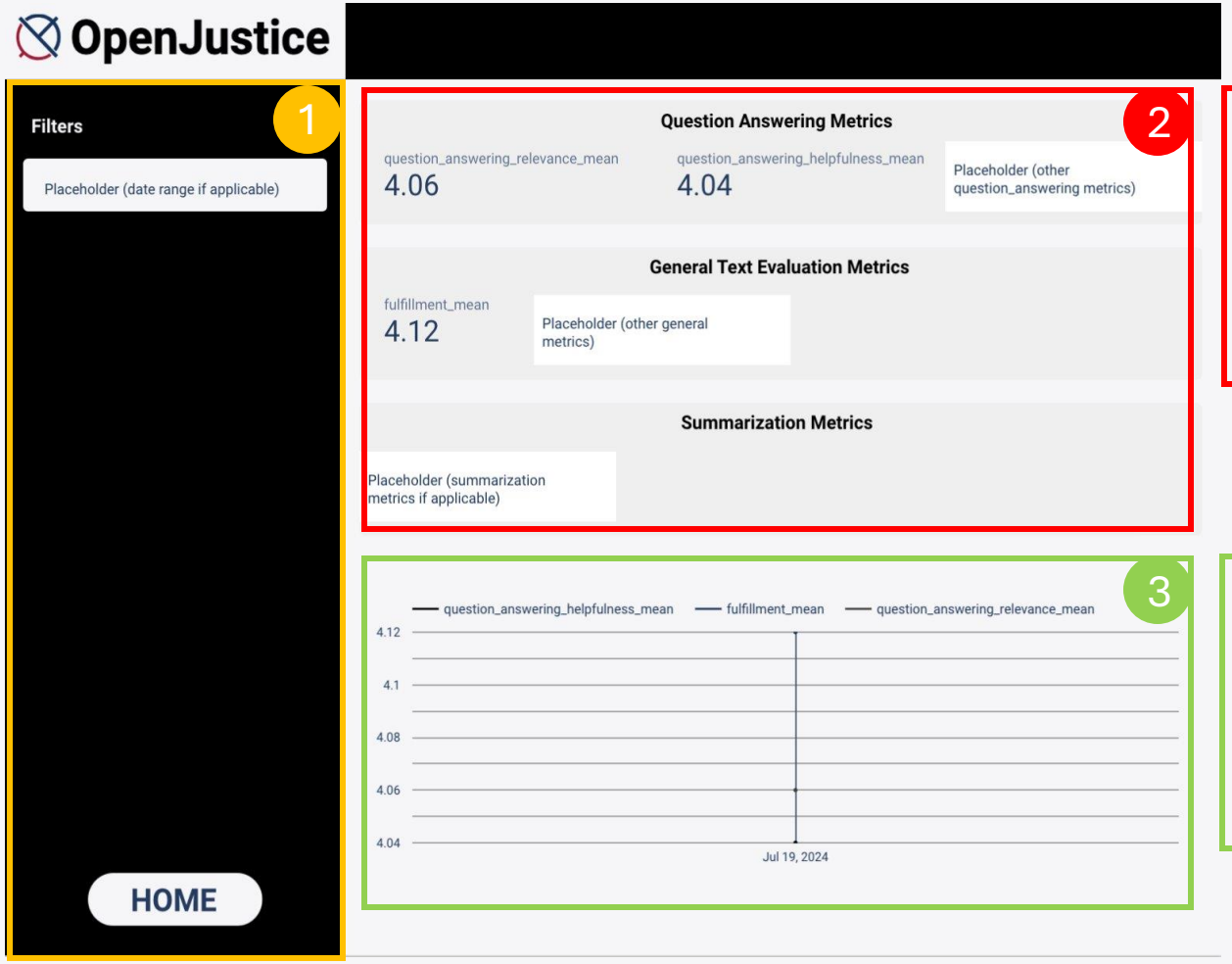
** A key event is an event that measures an action that's particularly important to the success of business



OpenJustice

R - Page 3 Overview

Filters:
Since it is a starting point for evaluating the conversation data, this proof-of-concept dashboard doesn't have any applicable filters yet, but later implementation is available.

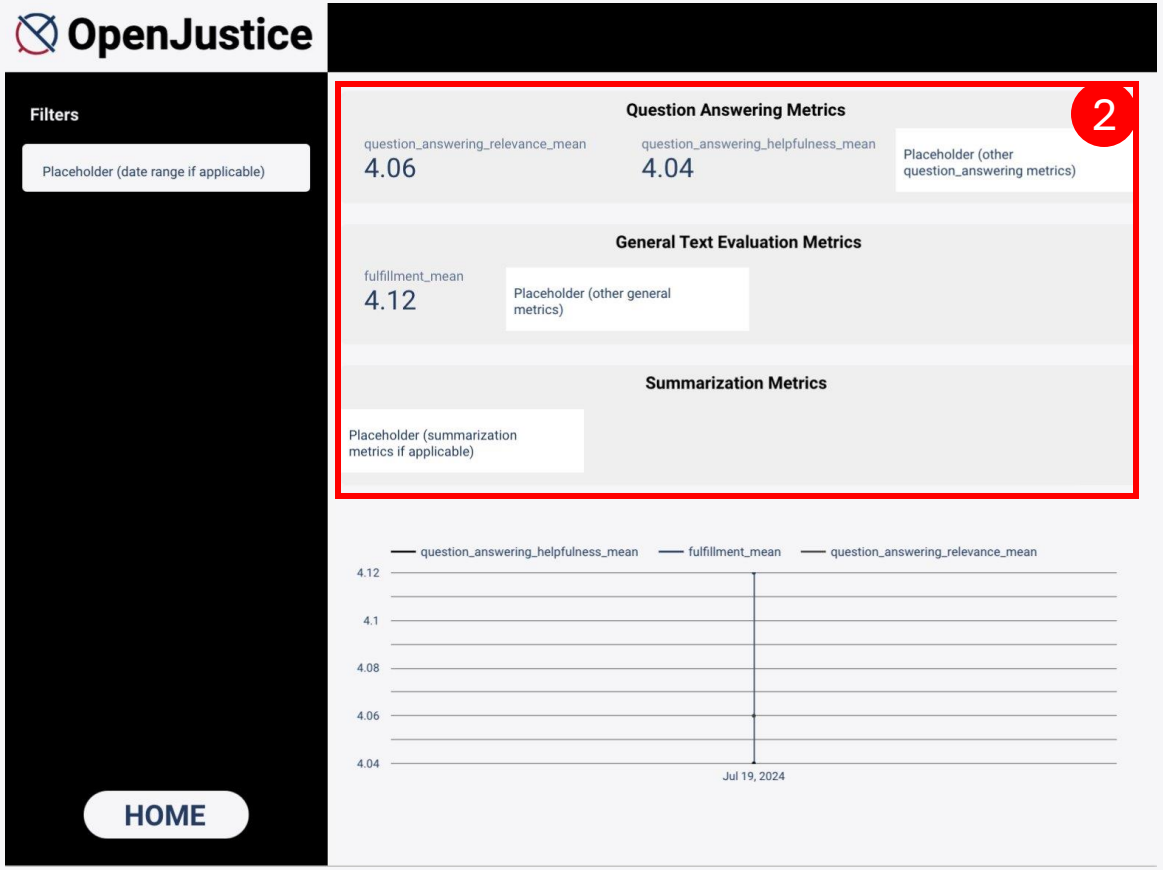


Scorecards:
Important metrics from evaluation results, which display the average score. Placeholders are for later possible metrics to be added.

Line chart:
Allow users to see a trend of evaluation scores changing over time, only one evaluation is included for now as a starting point

OpenJustice

S - Page 3 Scorecards

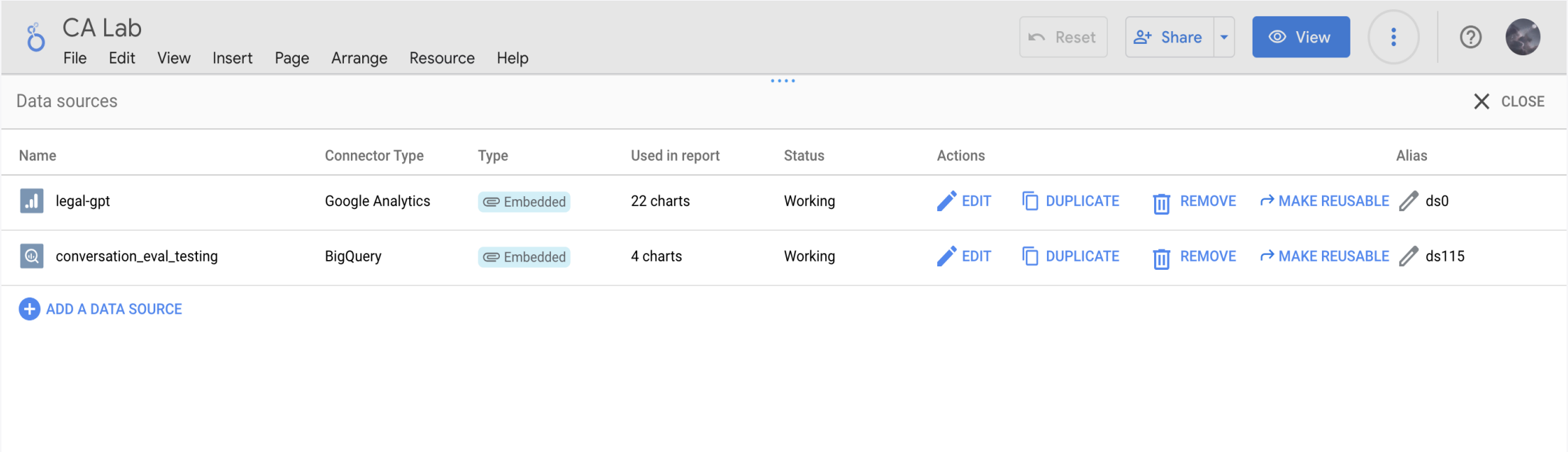


Metrics (Vertex AI default): there are the metrics that can be implemented now or in the future, more data features needed

1. Question answering (main focus): to evaluate the model's ability to answer questions (partially available now)
   - question_answering_quality
   - QuestionAnsweringHelpfulness
   - QuestionAnsweringCorrectness
   - QuestionAnsweringRelevance
2. General text generation: to evaluate the model's ability to ensure the responses are useful, safe, and effective for your users (partially available now)
   - Bleu
   - Rouge
   - Coherence
   - Fluency
   - Groundedness
   - Fulfillment
3. Summarization: to evaluate model summarization (NOT applicable for now)
   - summarization_quality
   - summarization_helpfulness
   - summarization_verbosity

OpenJustice

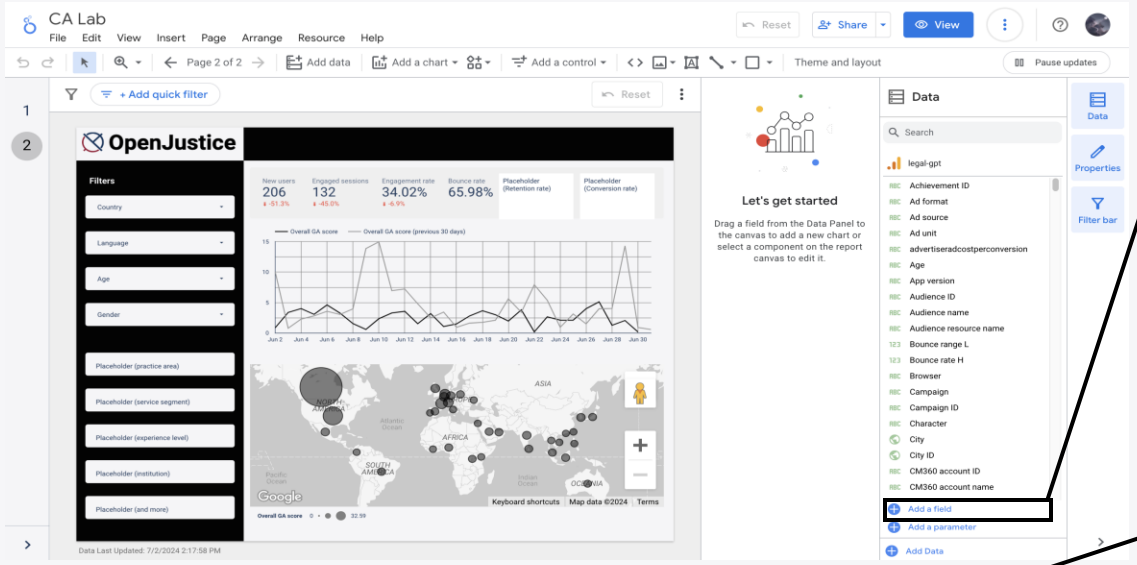
T - Different Data Sources



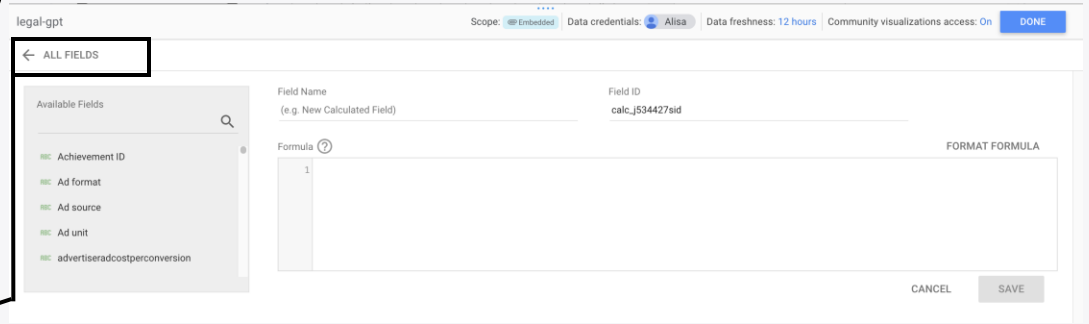
1. Resource -> Manage added data sources
2. This page will show the data sources that are connected to the dashboard (all data displayed are real-time values)
3. The BigQuery data can be modified from the notebook in Vertex AI:
    - Google Cloud -> Search "Vertex AI" -> workbench -> Open instance (START) -> Open Jupyter Notebook -> Run the notebook and a new row of evaluation results of all the conversation data in Firestore will be added with a new timestamp (Vertex AI, Firestore, and BigQuery are already initialized in the notebook)

OpenJustice

U - Creating New Field (e.g. Overall GA score)

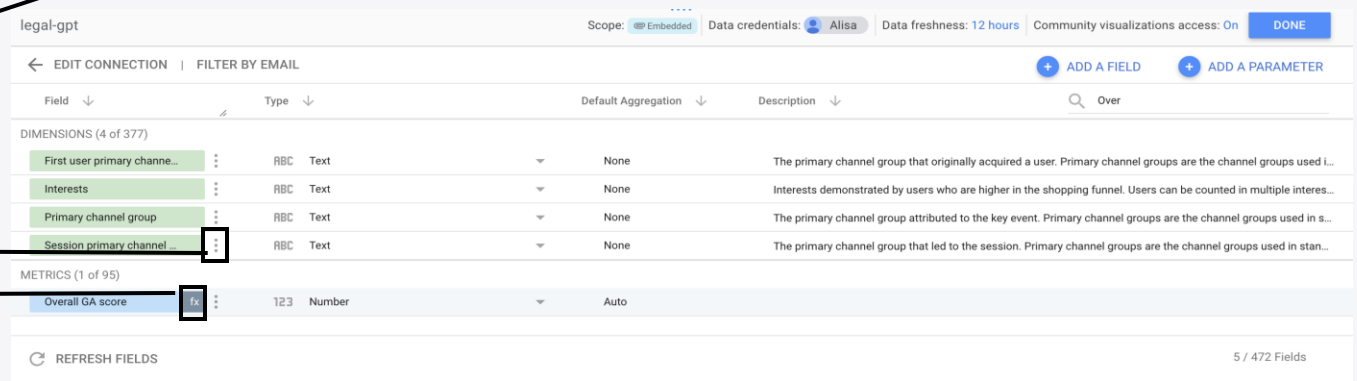


On the right where displays the data source:
Add a field -> Add calculated field
After filling the Formula, click on SAVE and then DONE



To modify, delete a Field:

1. Click on ALL FIELDS
2. Search the Field we want to change
3. Click the dots on the right to remove it
4. Click on the fx to modify the formula



OpenJustice