DOWNLOAD THE DATASETS

Full dataset:
https://drive.google.com/file/d/1fdpvAp3u7hksi0q-E4eJhAFPJogkur3z/view?usp=sharing
Sample dataset:
https://drive.google.com/file/d/1EzbSfRFda8rYjluxjlD-uBF-rjvMabUp/view?usp=sharing

data spec

Type of data that will be used for the representation.
Database 1: posts_questions
https://www.kaggle.com/datasets/stackoverflow/stackoverflow/data?select=posts_questions
id – int
title – string
body – string
accepted_answer_id - int
answer_count – int
comment_count – int
community_owned_date - Datetime
creation_date – Datetime
favorite_count - int
last_activity_date – Datetime
last_edit_date - Datetime
last_editor_display_name - string
last_editor_user_id - int
owner_display_name - string
owner_user_id – int
score – int
tags – string
view_count – int
parent_id - int
post_type_id - int

Database 2: Most Popular Programming Languages Since 2004
https://www.kaggle.com/datasets/muhammadkhalid/most-popular-programming-languages-since-2004
Date - Datetime
Abap - float (percentage)
Ada - float (percentage)
C/C++ - float (percentage)

C# - float (percentage)
Cobol - float (percentage)
Dart - float (percentage)
Delphi/Pascal - float (percentage)
Go - float (percentage)
Groovy - float (percentage)
Haskell - float (percentage)
Java - float (percentage)
JavaScript - float (percentage)
Julia - float (percentage)
Kotlin - float (percentage)
Lua - float (percentage)
Matlab - float (percentage)
Objective-C - float (percentage)
Perl - float (percentage)
PHP - float (percentage)
Powershell - float (percentage)
Python - float (percentage)
R - float (percentage)
Ruby - float (percentage)
Rust - float (percentage)
Scala - float (percentage)
Swift - float (percentage)
TypeScript - float (percentage)
VBA - float (percentage)
Visual Basic - float (percentage)

Database 3: GitHub Programming Languages Data
https://www.kaggle.com/datasets/isaacwen/github-programming-languages-data
name - string
**Language name
year - Datetime
quarter - int
Count - int
**Number of issues on repos with the corresponding language

Default value
We don't plan to set up default values for any fields. Missing values will be marked with N/A, essential fields such as id and owner_user_id will be enforced, and entries will be removed if those are missing.

Range of value
Database 1: posts_questions
id – [4, 73842327]

title (text)
body (text)
accepted_answer_id [7, 73842204]
answer_count – [0, 518]
comment_count – [0, 108]
community_owned_date - [07/31/2008 21:42:52, 03/24/2022 03:09:30]
creation_date – [07/31/2008 21:42:52, 09/25/2022 05:56:32]
favorite_count - [0, 11649]
last_activity_date – [09/04/2008 12:50:25, 09/25/2022 05:56:36]
last_edit_date - [07/31/2008 21:42:52, 09/25/2022 05:50:31]
last_editor_display_name - (text)
last_editor_user_id - [1, 20061844]
owner_display_name - (text)
owner_user_id – [1, 20081043]
score – [-146, 26621]
tags (text)
view_count – [1, 11649204]
parent_id - (empty for all)
post_type_id - (1 for all)

Database 2: Most Popular Programming Languages Since 2004
Date - [2004-06-30, 2024-11-30]
**Any Language - [0, 1] (since it's a percentage)

Database 3: GitHub Programming Languages Data
name - (text)
year - [2011, 2022]
quarter - [1, 4]
Count - [100, 341 000]

Simplified analysis of the distribution of values
All of our data will be considered in the timeframe 2011-2022 to accommodate all of our datasets. This is also a long enough timeframe to make an interesting analysis about programming practices and sentiment of the users regarding the most popular programming languages.
The entries will be picked randomly and uniformly per year. That is we will have the same number of entries for every year from 2011 to 2022 (~100 entries per year, giving ~1100 entries total). Months, days, and specific times will not be considered, only years.
Int values range from 0 to no more than 80,000,000.

Are these values unique?
In general, every id value, such as id and owner_user_id, is unique. The rest of the values are either text values (String) that may not be unique (for example, the body or the title of the post),

or numeric values (int, float) that could be repeated across the database (view_count, score, etc.).

Will you use this value (maybe in composition with others) to detect possible duplicate records? If so, how?
We will use 2 of our unique values, owner_user_id and id, to detect duplicates. If there already exists an entry with the same owner_user_id and id, the duplicate values will be removed to avoid redundancy.

Is this a required value?
Yes, both owner_user_id and id are required values. They are both necessary to check for uniqueness of entries in our database. If the values in the database are missing, the entry will be deleted.

Do you plan to use this attribute/feature in the analysis? If so, how?
Database 1: posts_questions
id – Used for unique identification of the post on StackOverflow.
title – Used for analysis of vocabulary associated with different sentiments.
body – Used for analysis of vocabulary associated with different sentiments.
accepted_answer_id - NOT USED
answer_count – Used to measure interest / relevance of the post.
community_owned_date - NOT USED
comment_count – Used to measure interest / relevance of the post.
creation_date – Used for sentiment analysis over time / interest on the topic over time.
favorite_count - NOT USED
last_activity_date – Used for sentiment analysis over time / interest on the topic over time.
last_edit_date - NOT USED
last_editor_display_name - NOT USED
last_editor_user_id - NOT USED
owner_display_name - NOT USED
owner_user_id – Used for unique identification of the post on StackOverflow.
score – Used to measure interest / relevance of the post.
tags – Used to identify which programming languages are associated with which sentiments.
view_count – Used to measure interest / relevance of the post.
parent_id - NOT USED
post_type_id - NOT USED

Database 2: Most Popular Programming Languages Since 2004
Date - Used to match each year from 2011 to 2022 to the programming language corresponding to the highest number of issues reported on Github.
**All Programming Language – All languages will be initially included to find the ones that were causing the most Github issues per year from 2011 to 2022.

Database 3: GitHub Programming Languages Data

name - Name of the most popular programming language. Used to match each year from 2011 to 2022 with its most popular programming language.
year - Used to match each year from 2011 to 2022 with its most popular programming language.
quarter - NOT USED
Count - Used to match each year from 2011 to 2022 with its most popular programming language (by number of counts!).

Does this feature include potentially sensitive information? If so, how do you suggest handling such issues?
Entries with identifying names will not be used in the analysis, therefore, none of the data used directly contains sensitive information. All posts and owner_user_id have been anonymized through ids. Nevertheless, the titles and content could be manually searched online and the identities of the users could be revealed. This would not be too serious of a security issue since the information is publicly available anyways. Therefore, no further anonymization will be necessary.