Big Data Platforms Final Project

Will TuringBots replace human software developers?

Kenny Wang

Executive Summary

Background

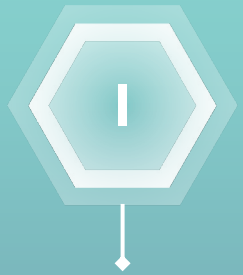
The project investigates if AI tools like GitHub and similar assistants can significantly improve developer productivity and replace human software engineers. Using GitHub Archive data stored in Google Cloud, we analyzed trends, patterns, and data correlations across repositories, programming languages, licenses, and commits.

Key findings

- **Data Timeline:** The data collection gaps and spikes in activity align with significant tech improvements. It growth rapidly start around 2007, and decreased around 2018.
- **Programming Languages:** JavaScript, CSS, HTML, Shell, and Python are the most popular programming languages on GitHub, since these languages are essential for Data Science/AI related projects.
- **Commit Reasons:** Major reasons for commits: New_devolopment, bug Fixes, and testing are the main reasons for commits which represent the continuous improvements for human developer.
- **Licenses:** MIT License dominates GitHub repositories, especially for projects with JavaScript, HTML and CSS.
- **Text Duplication:** 95%+ of commit messages are unique, indicating diverse developer input.
- **Prolific Committers / AI and Human Collaboration:** Auto Systems and Other individuals are the main committers to the commits volumes. In other words, AI tools significantly enhance developer productivity, automating routine tasks and accelerating innovation. However, human creativity and oversight remain critical for complex problem-solving.

TABLE OF CONTENTS

Methodology and Data Overview



Pre-Processing

- Removed erroneous, incomplete, and duplicate records using Spark
- Extracted key variables from deeply nested fields in the parquet files.
- Sampled data for initial exploratory analysis and refined methods before scaling.

- Identified key variables
- Examined trends
- Make visualizations
- Data Transformation

Exploratory Data Analysis

Key Analyses

- Programming Language and License analysis
- Most popular technology and repositories analysis
- Subject and Message uniqueness analysis
- Timeline analysis

- Spark: Used for distributed data processing and aggregation
- Python Libraries: Pandas, PySpark, etc. for advanced analytics; Matplotlib and Seaborn for visualizations

Tools and Technologies

Methodology and Data Overview

Data Size: ~ 1.36TiB

Data Source:
GitHub Archive that is stored in Google Cloud Storage
gs://msca-bdp-data-open/final_project_git folder

Sub-folder	Variables	Info
Commits	commit, tree, parent, author, committer, subject, message, trailer, difference, difference_truncated, repo_name, encoding	This contains information about the commits made to repositories.
Contents	id, size, content, binary, copied	Provides the content of the files in the repositories.
Files	repo_name, ref, path, mode, id, symlink_target	This contains metadata about the files in the repositories.
Languages	repo_name, language	This table provides an aggregation of the number of bytes of code for each language in a repository.
Licenses	repo_name, license	Contains information on the licenses used by repositories.

Data Cleaning

Datasets

Commits
- Extracted key fields from nested arrays, author and committer.
- Converted to timestamp; extracted array repo_name to string.
- Excluded future dates and trivial messages.
- **Remained Variables:** commit, author_name, author_email, author_date, committer_name, committer_email, committer_date, subject, message, repo_name

Licenses
- **Remained Variables:** repo_name, license

Files
- Dropped irrelevant null columns, symlink_target
- **Remained Variables**: repo_name, ref, path, mode, id

Contents
- Filtered non-binary content with size between 100 bytes and 1 MB
- **Remained Variables**: repo_name, size, content, binary, copies

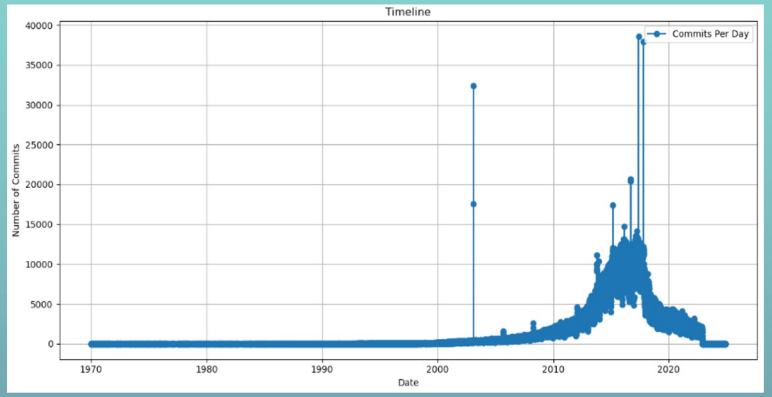
Languages
- Exploded nested arrays, language
- **Remained Variables:** repo_name, language, bytes

Steps taken

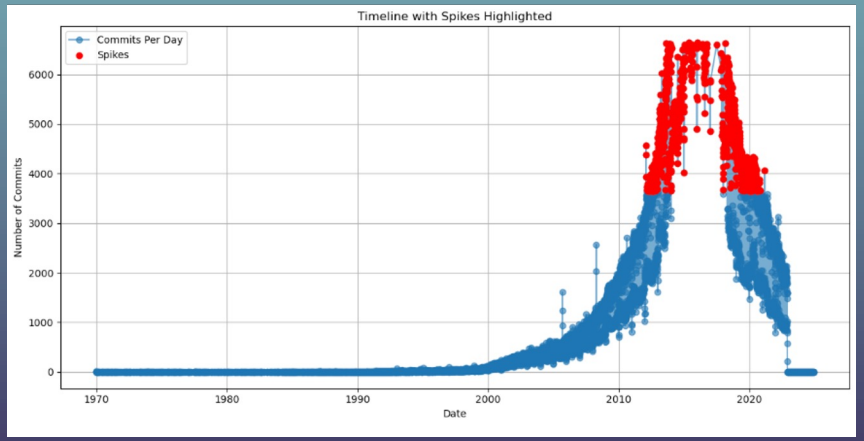
1. Sampling: Applied a 10% sample for testing and optimization.
2. Removed duplicate records across all datasets.
3. Flattening Nested Structures: Used explode for nested fields in the Languages and Commits dataset.
4. Filtering to ensure valid and non-null entries.
5. Cleaned datasets saved as Parquet files to Google Cloud Storage for optimized processing. Path is: gs://msca-bdp-students-bucket/notebooks/jingkaiw/cleaned_data

Timeline Analysis

Original Graph



Graph after removing outliers with spikes highlighted



Timeline of the data
The timeline start before 1970, and extends to 2024. The most of the activity mainly focus between 2008 to 2018.

Peaks and valleys
For the significant peaks, it is during 2015-2017, which exceed 30,000 commits. And as for the valleys, the significant valleys are very obvious at the timeline before 2000, the early timeline, and towards the end, after around 2018.

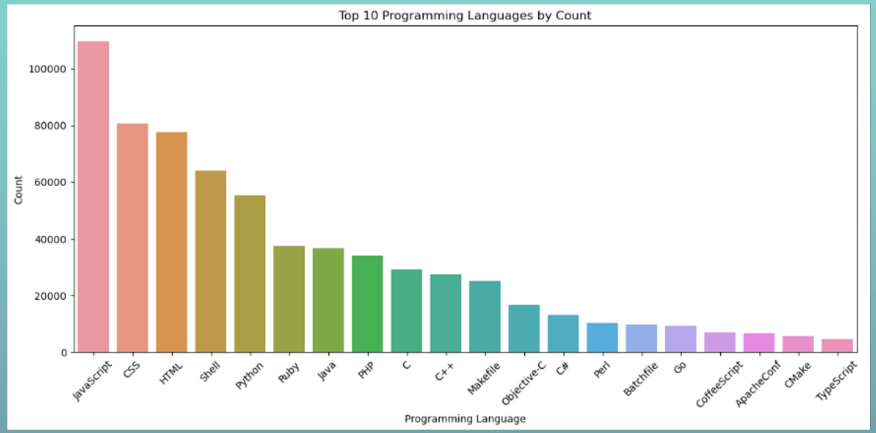
Data collection gaps
For the data collection before 2007, it indicates that the limited GitHub usage or incomplete data recording. Besides, after ~2018, a sharp decline in commits, which may because of the reduction in activity or gaps in the dataset.

Outliers
There are clear outliers in the earlier years, such as unusually high spikes in commit counts, likely due to errors or data aggregation issues.

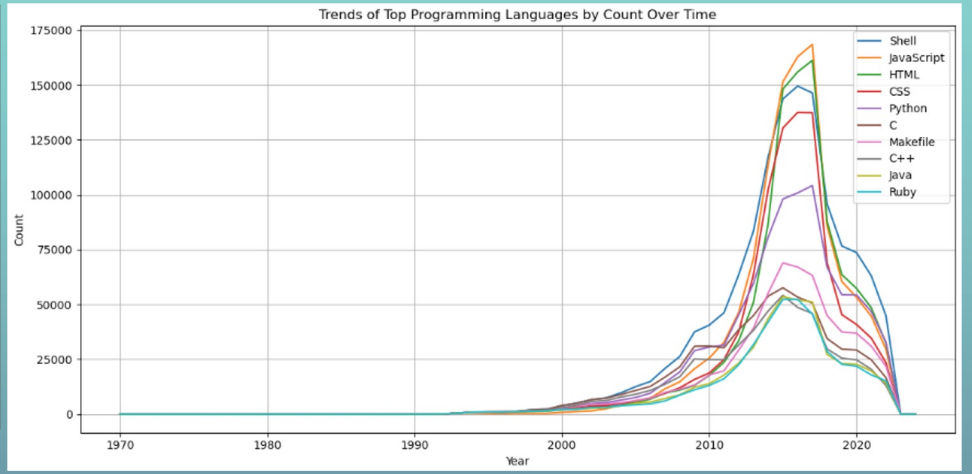
Spikes
Several spikes are evident, particularly in the peak years. Most spikes appear to be caused by real activities, like collaborative development on popular repositories. And Some spikes in earlier years, may result from data inconsistencies or automated processes.

Programming Language and Licenses Analysis

Popularity graph



Trends graph



The most popular programming languages on GitHub

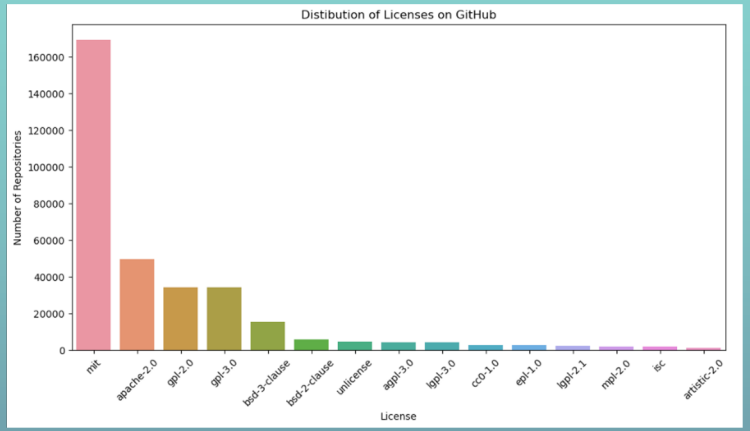
JavaScript, CSS, HTML, Shell, and Python are the most 5 popular programming languages on GitHub. For the first three, they form the foundation of web development, an area with massive growth and a large number of projects. Besides, Shell scripting is widely used for backend automation. In addition, Python is indispensable for a broad range of domains, including data science and machine learning. Their combined utility, widespread applicability, and strong community support drive their popularity.

The trend of most popular programming languages change over time

Ruby maintains consistent popularity until ~2015, and JavaScript and HTML exceed it. The trends highlight the evolution of technology and developer needs. Web and data-driven applications caused a surge in JavaScript, HTML, CSS, and Python, while legacy languages like C, C++, and Java remained essential but less dominant.

Programming Language and Licenses Analysis

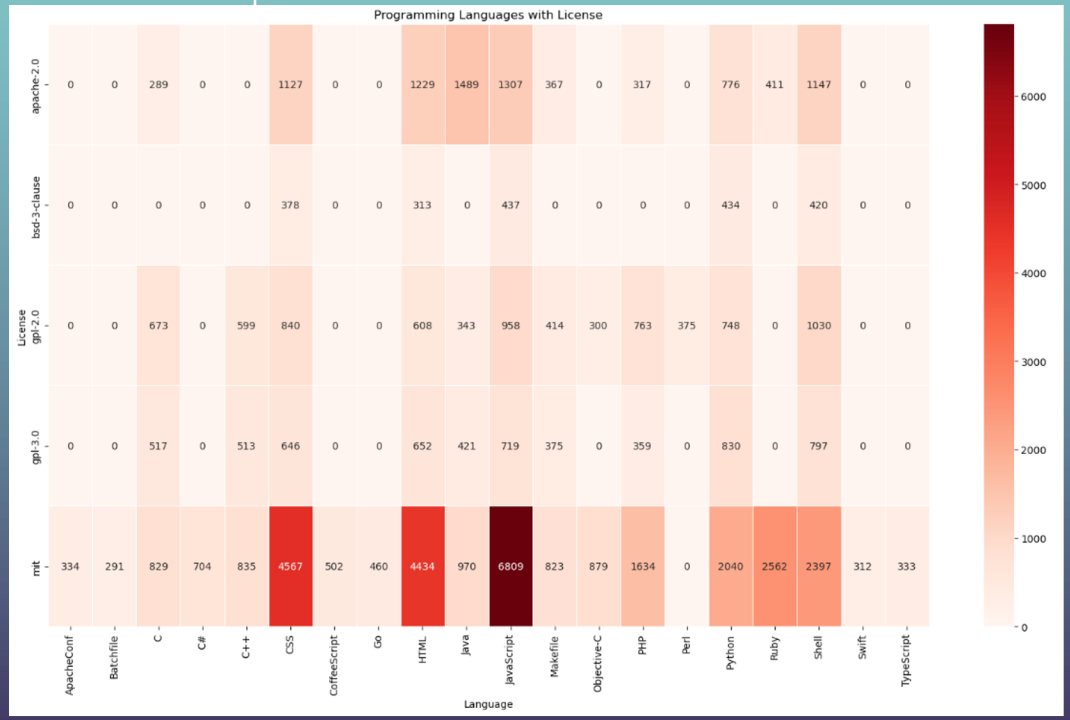
Licenses distribution



Distribution of licenses across GitHub repositories

MIT license is the most widely used due to the flexibility, developer-friendly terms, and legal protections. Rest, Apache 2.0 is popular since it provide a robust legal framework. Besides, copyleft licenses like GPL 2.0 and 3.0 remain popular for ensuring collaborative projects remain open-source and protecting contributors' rights.

Association Heatmap

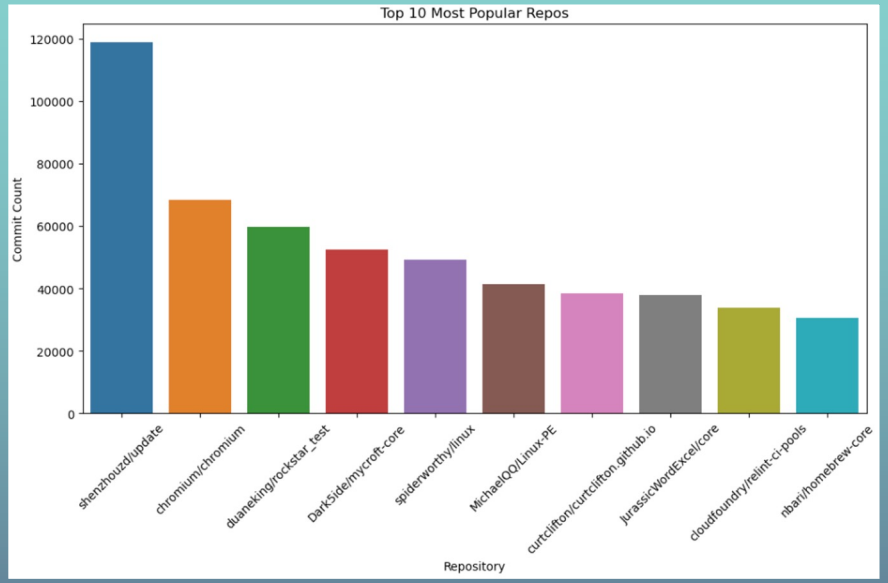


Programming languages associated with licenses

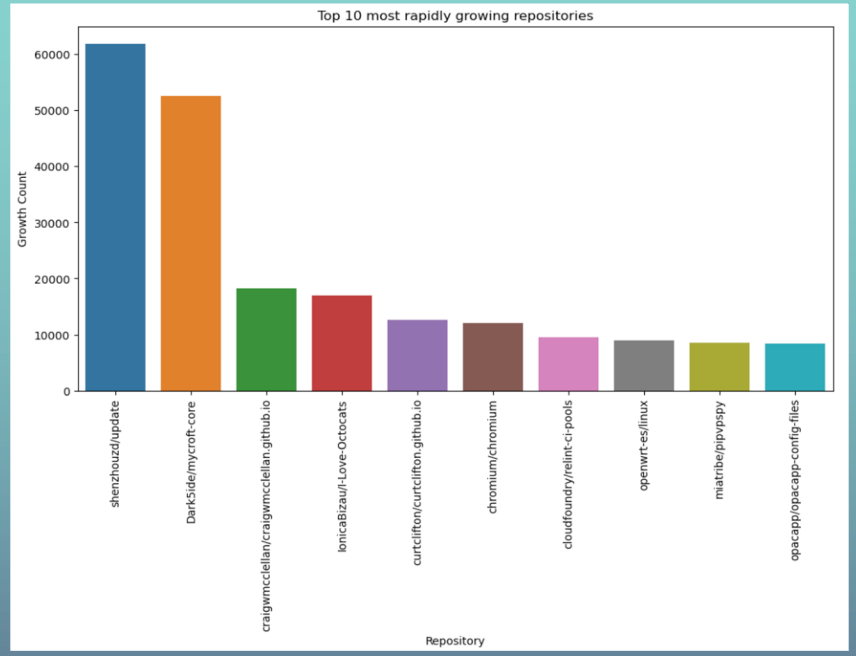
- **MIT License** associate with JavaScript the most. It is widely associated with popular languages like JavaScript, Python, HTML, and CSS due to its permissiveness and minimal restrictions.
- **GPL-3.0 and GPL-2.0 Licenses** are more prevalent in languages like C and C++, for valuing strong copyleft protections to remain open-source.
- **Apache-2.0 License** is popular among projects written in Java, particularly in enterprise-level as it offers strong legal protection.

Most popular technology and repositories analysis

Popularity Repos graph



Rapidly Growing Repos graph



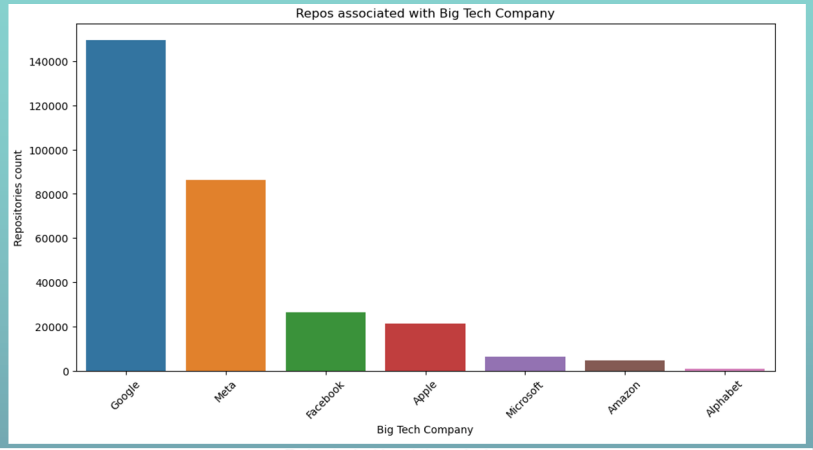
The most popular and rapidly growing repositories

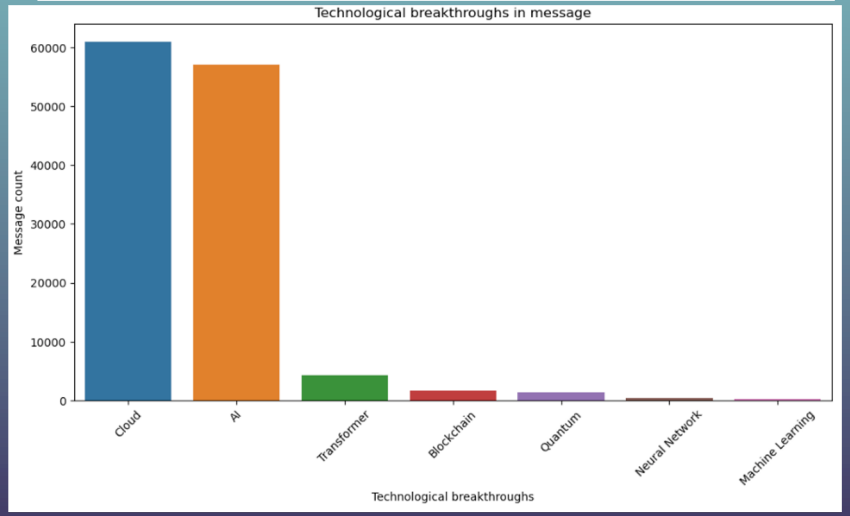
- The most popular repository, **shenzhouzd/update**, leads in both popularity, 118,733 commits, and rapid growth metrics.
- **chromium/chromium** reflect community-driven open-source projects with active contributions from thousands of developers
- Rapid growth repositories like **Dark5ide/mycroft-core** show signs of active community engagement.

Technology driving popularity or explosive growth

- Web and browser technology development with JavaScript, CSS. Repos like chromium highlights the persistent demand for browser technology.
- AI and voice technology with Python become more popular, so repositories like Dark5ide/mycroft-core growing rapidly.
- Technologies catering to cloud computing and DevOps workflows, which are experiencing steady growth as businesses scale operations digitally.

Most popular technology and repositories analysis





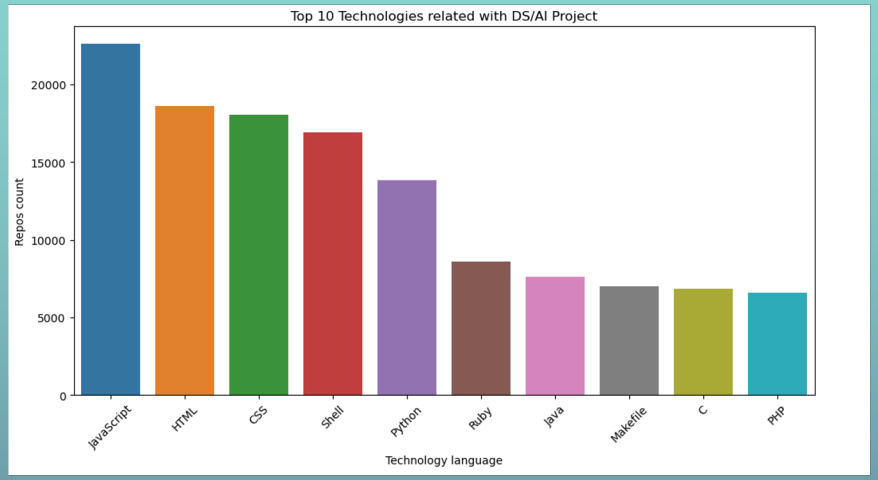
Association with Big Tech and Open Sourcing

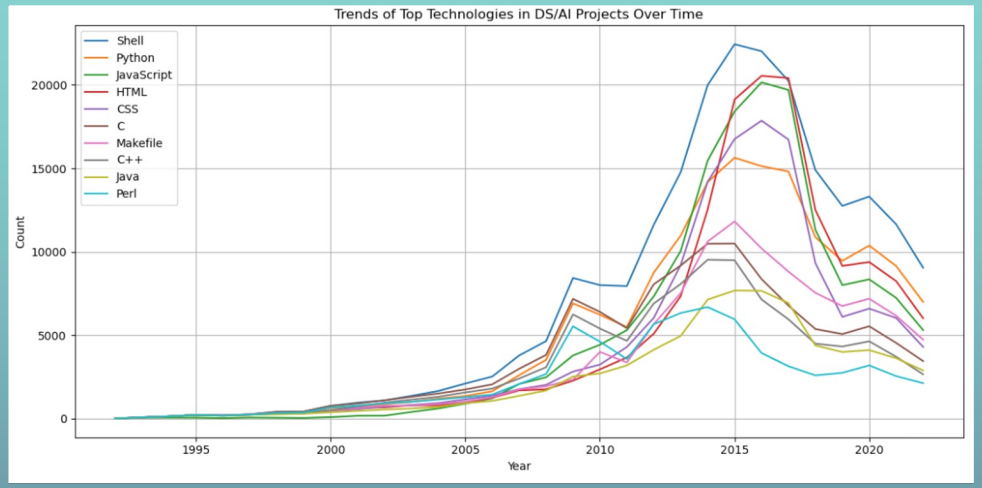
The top repositories are closely associated with Big Tech companies like Google, Meta, Apple, and Microsoft. The above chart illustrates how Google and Meta lead in the open-source ecosystem, showcasing the massive number of repositories under their stewardship.

Technological Breakthroughs Driving Adoption

- Cloud Computing: With tools like Kubernetes for Google and AWS for Amazon, Big Tech has transformed how developers build scalable and flexible applications. Cloud computing underpins nearly all modern software development.
- AI: Frameworks like TensorFlow for Google and PyTorch for Meta are pivotal in AI and machine learning, enabling a broad range of applications from natural language processing to computer vision.

Data Science/AI Projects Analysis





Technologies Associated with Data Science or AI Projects

- JavaScript, with 20,000+ repositories, is the top technology for DS/AI projects because of the wide application in building front-end interfaces.
- HTML and CSS are crucial for creating web visualizations, dashboards, and front-end integrations in DS/AI projects.
- Shell: Often used for automation, deployment, and environment management in DS/AI workflows,.
- Python is the key language in AI/ML with its libraries like PyTorch, and scikit-learn.

Trends Over Time:

- Technologies like C and Java, lead before 2005 due to their use in foundational computing and early ML libraries.
- After 2005, Python experienced explosive growth, because of the simplicity, extensive AI/ML libraries, and web technologies.
- Shell maintain consistent related with DS/AI Project. HTML became more related and reach the peak until 2016.
- For JavaScript, HTML, CSS, the increasing demand for web-based AI applications and dashboards has cemented these as primary technologies in DS/AI projects.

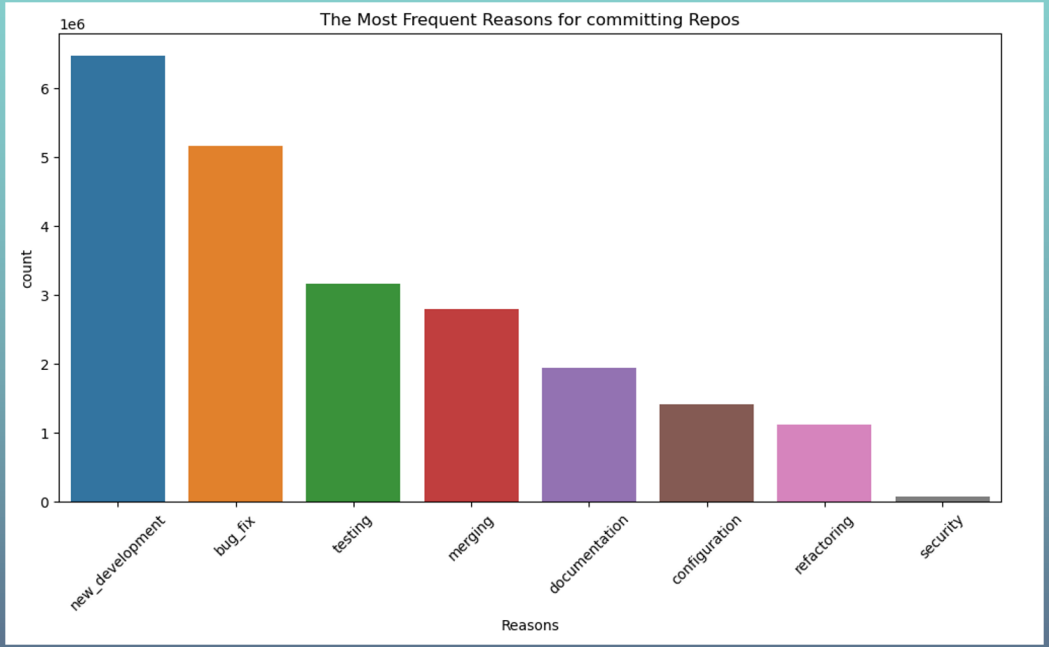
Commit Reasons Analysis

New Development is the most frequent reason for commits, with over 6 million instances. For this reason, it associated with terms like "feature," "add," "new," and "create." This reason emphasized the innovation and building new functionalities.

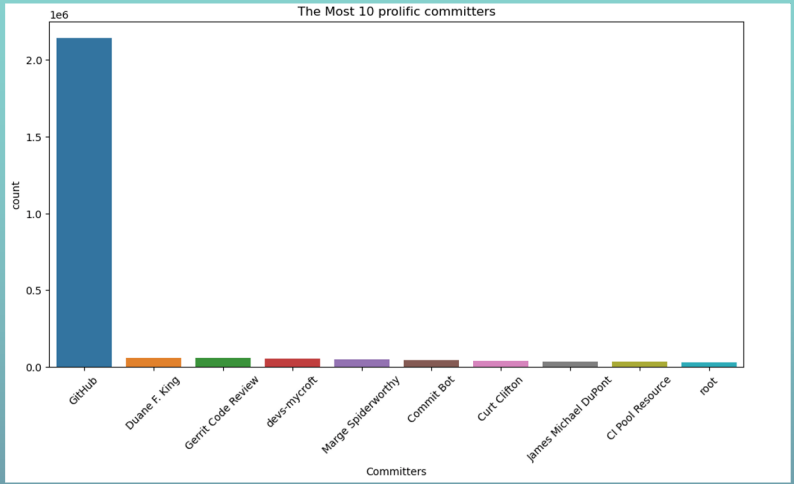
Bug Fixes, the second highest, reflecting the importance of maintaining stability and reliability in projects with "fix," "bug," "resolve" related.

Testing is a significant portion of commits involve testing-related activities. Words like "test," and "integration" are prevalent, which emphasize the critical role of quality assurance.

Merging suggests the active collaboration and frequent integration of work by developers.

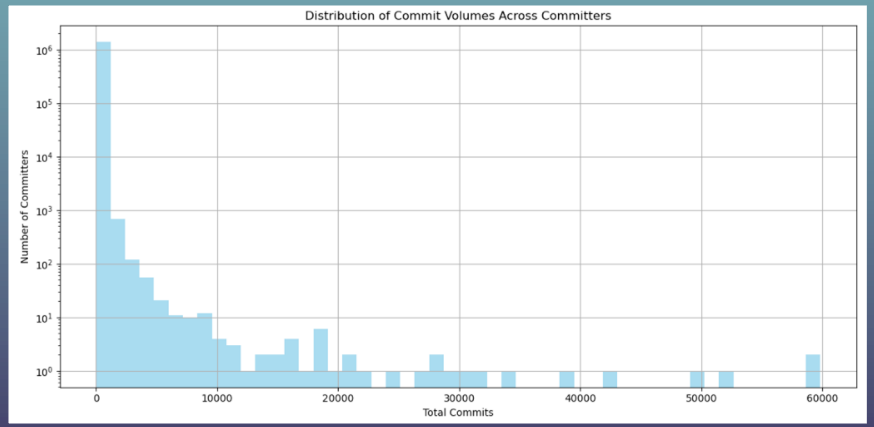


Prolific Committers Analysis



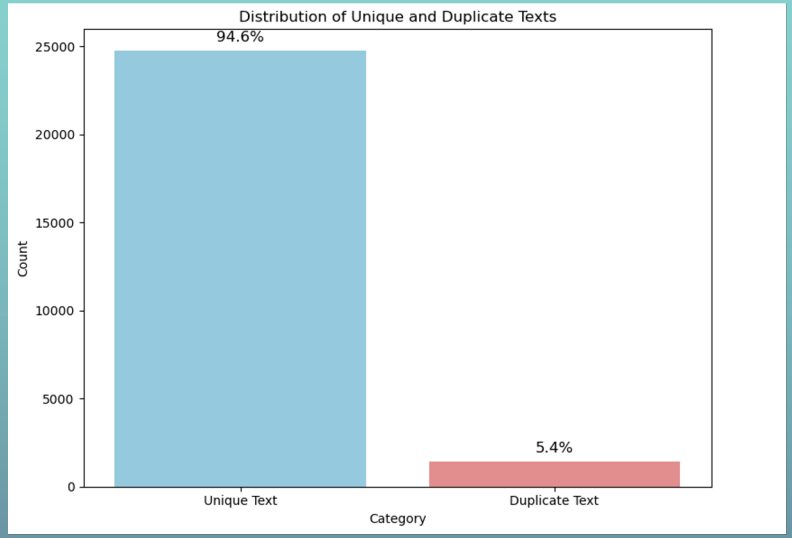
The Most Prolific/Influential Committers:

The most prolific committer is **GitHub**, which over 2 million commits. These are likely system-generated commits related to GitHub's platform operations. E.g. pull requests, merges. For the second, **Gerrit Code Review** is an tool to review code. And the rest individual contributors like **Duane E. King**, and **devs-mycroft** follow with significantly fewer but impactful contributions.

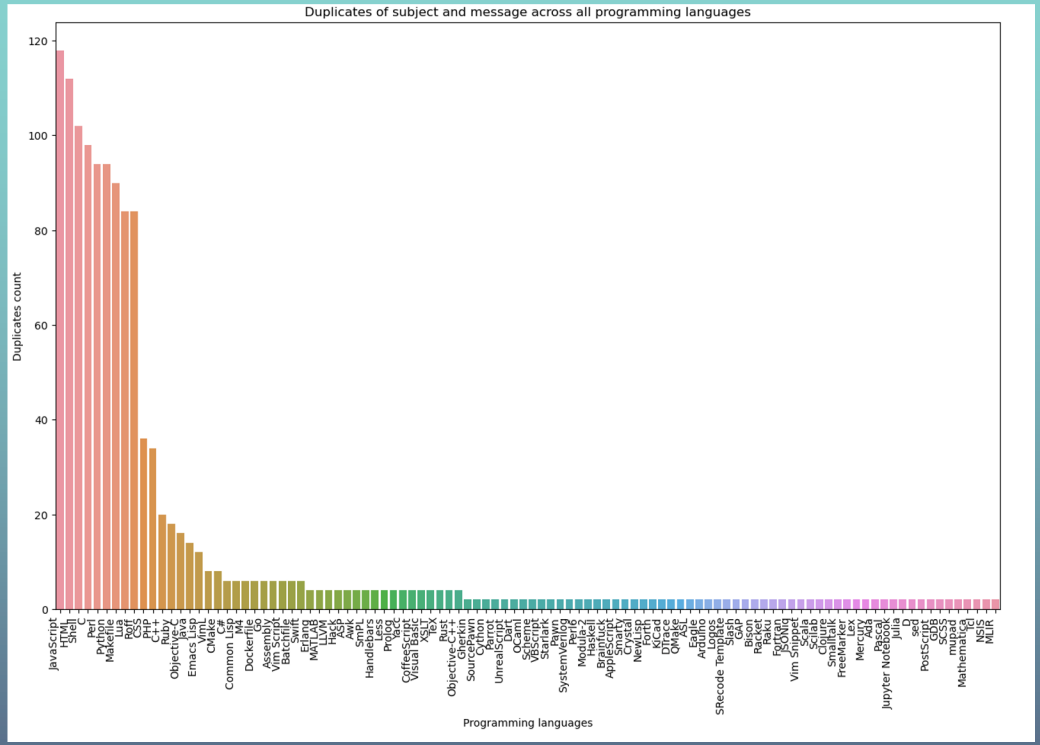


The histogram of commit volumes demonstrates a **heavy-tailed distribution.** The skewed distribution suggests that a few individuals drive the a lot of development activities.

Subject and Message uniqueness analysis

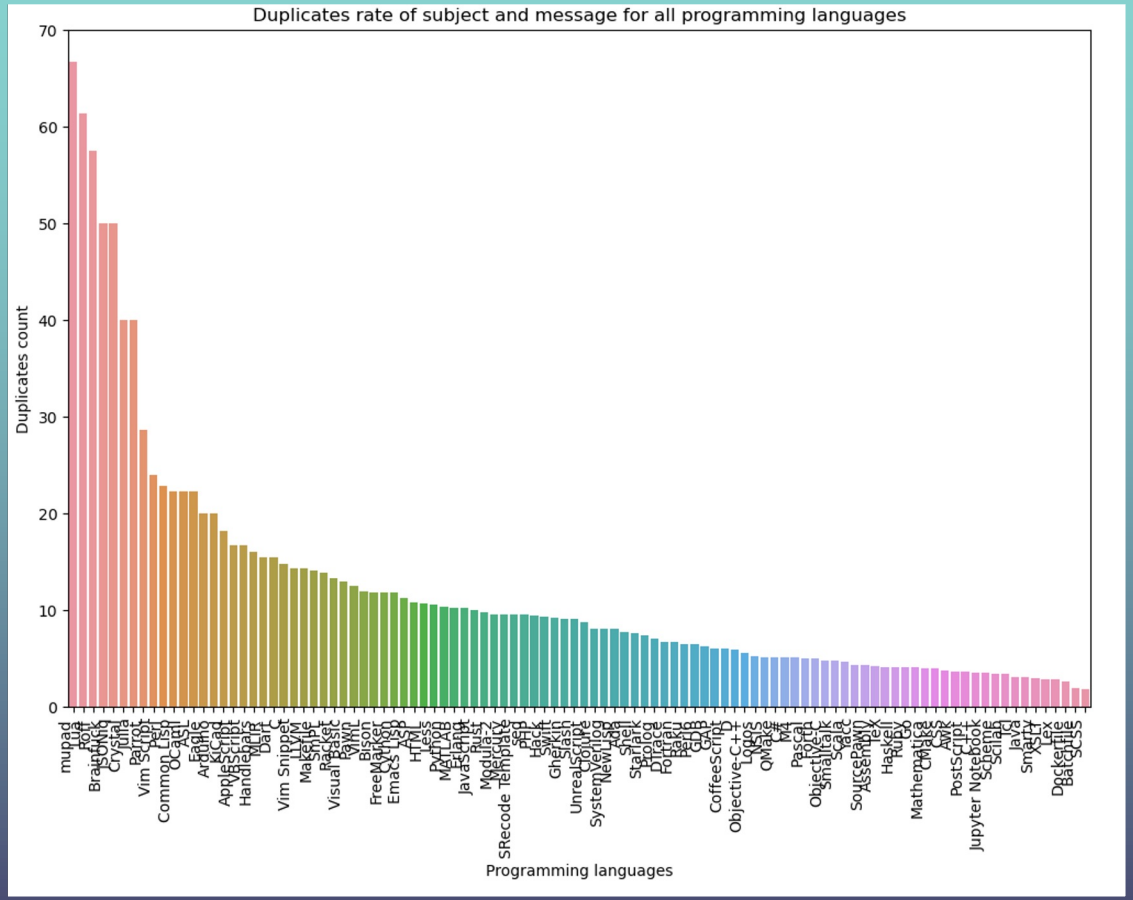


The majority of the texts are unique, accounting for **94.6%** of the dataset, while **5.4%** are duplicates. This means that GitHub repositories tend to have high variability in commit subjects and messages,.



JavaScript, HTML, Shell, C, and Perl have the highest number of duplicate commit messages. These languages are common across various types of projects, including web development, scripting, and data science.

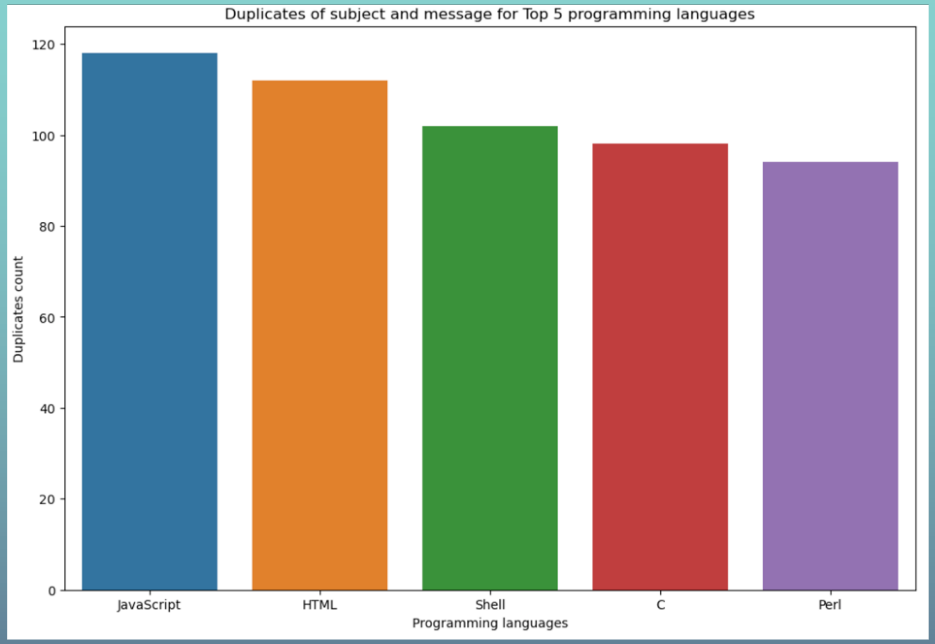
Subject and Message uniqueness analysis

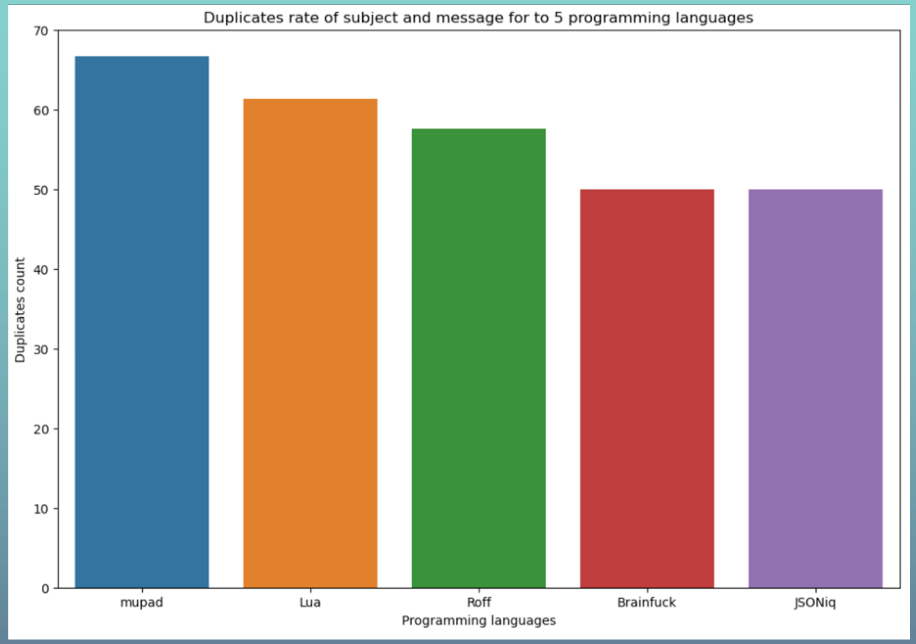


In this chart, normalized by duplication rate, languages like **mupad**, **Lua**, and **JSONiq** have higher duplication rates. This indicates that, relative to their total use, these languages have a significant proportion of duplicates.

Normalizing by duplication rate accounts for the total usage of each language. This removes the bias of large-scale popularity, making it possible to highlight languages where duplication is inherently more frequent.

Subject and Message uniqueness analysis





For Duplicate Counts: **JavaScript**, **HTML**, **Shell**, C, and **Perl** have the highest duplicate counts,which >=100, driven by their popularity and widespread use in diverse projects.

For Duplicate Rates: **Mupad**, **Lua**, **Roff**, **Brainfuck**, and **JSONiq** exhibit the highest duplicate rates, with all 50%+, indicating repetitive and template-based commit patterns.

Conclusions and actionable recommendations

Conclusions

1. JavaScript, CSS, and HTML are the most popular programming languages on GitHub due to their essential role in Data Science, AI, and web development.
2. The MIT license is the most popular due to its simplicity and permissiveness, and it has a strong association with JavaScript, HTML, CSS and Python.
3. Popular and rapidly growing repositories, like **shenzhouzd/update,** are often driven by Big Tech or cutting-edge technologies like AI and cloud.
4. AI/Data Science projects heavily use Python, JavaScript, and Shell, showing steady growth over time.
5. New development and bug fixes are the most frequent reasons for commits.
6. GitHub bots highly lead commit volumes, but human contributions show a diverse activity distribution.
7. Most commit messages are unique (~95%), with duplication concentrated in templated or repetitive tasks.

Recommendations

1. Find data collection gaps with fit solution to ensure the continuous and consistent recording the data,
2. Continuous improve popular languages with more convenient tools to simplify workflows like Python and JavaScript to meet industry trends.
3. Promote permissive licenses,like MIT License, to encourage contributions while ensuring legal compliance where needed.
4. Support rapidly growing repositories with funding and contributor onboarding.
5. Enhance automation for repetitive tasks and provide clear commit guidelines.
6. Investigate duplication trends in JSON and HTML to optimize project structures.
7. Leverage Big Tech open-source technologies and foster partnerships for innovation.
8. Automate commit message templates to reduce duplication and improve clarity.