BDP Final Project

Presented by Ursula Guo

Mind map

01 Executive Summary

Methodology and source data overview

Data clean-up and filtering

02 Exploratory Data Analysis

03 Timeline analysis

04 Programming Language and License Analysis

05 Most Popular Technology and Repositories

06 Subject and Message Uniqueness

07 Conclusion And Recommendations

Presentation 2024

Executive Summary

The analysis highlights trends in programming languages, licensing, and the most active repositories. Key findings include:

- Programming Language Trends: Shell, Python, and C are among the most historically significant languages by commit count, with Makefile gaining prominence post-2013.
- Licensing Patterns: The MIT license is the most widely used, showing strong correlations with JavaScript, HTML, and CSS.
- Repository Insights: AI-related repositories dominate commits, with natural language processing (NLP) leading AI technology trends.
- Commit Reasons: Feature development is the top reason for commits, comprising over 34% of the total.

This report identifies patterns of adoption and activity within the GitHub ecosystem and surfaces opportunities for further exploration in repository development and AI technology focus areas.

Methodology and Source Data Overview

Three tables were used in the whole of the analysis.

- Commits: contains information on the commit history of GitHub public repositories.
- Languages: contains information about programming languages used in each repository.
- Licenses: contains information on the licenses used by each repository.

Two other data sources were also provided to us (Contents and Files), but because they contain granular code-related data that is less relevant to the scope of our analysis, I did not use them.

The main methodology is to use Spark to query data, then converting to Pandas data frame for graphing.

Presentation 2024

-

Data clean-up and filtering

The committing time has a cutoff, after which all the dates have only round 2 commit counts, which is unrealistic. I probed around and found that after 2022-11-27, the data started looking odd. So I filtered out rows in the commit table if they have dates after 2022-11-27.

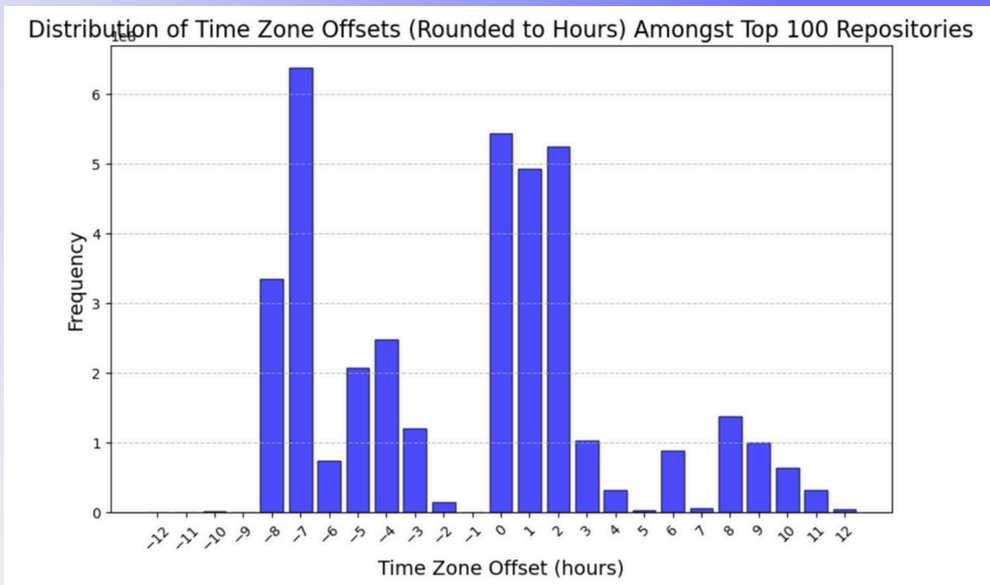
```
committer_timeline_pdDf.tail(5560)
```

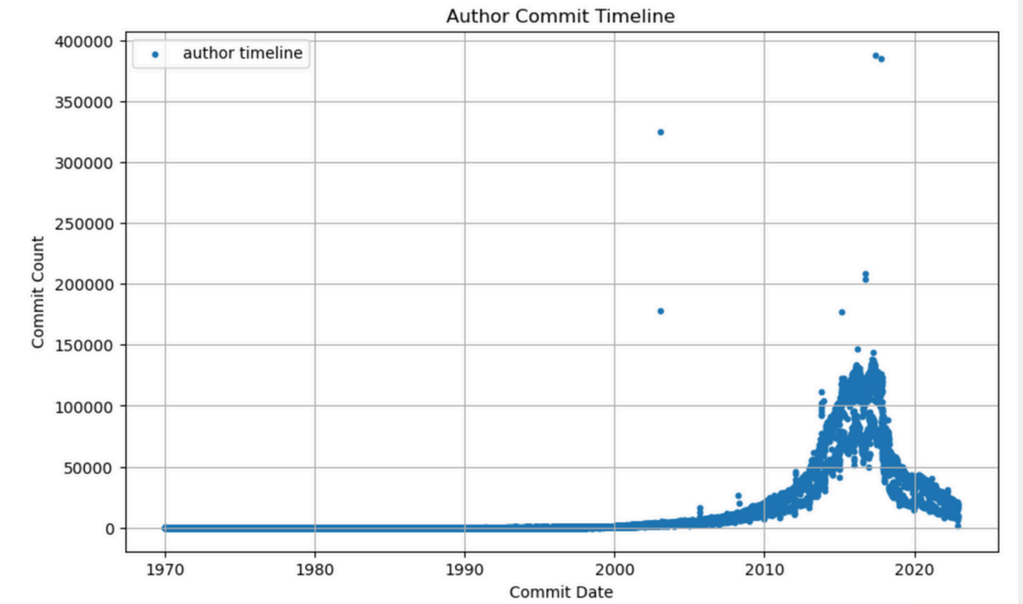
	committer_commit_date	committer_commit_count	Date
19322	2022-11-25	6168	2022-11-25
19323	2022-11-26	2840	2022-11-26
19324	2022-11-27	2	2022-11-27
19325	2022-11-28	2	2022-11-28
19326	2022-11-29	2	2022-11-29

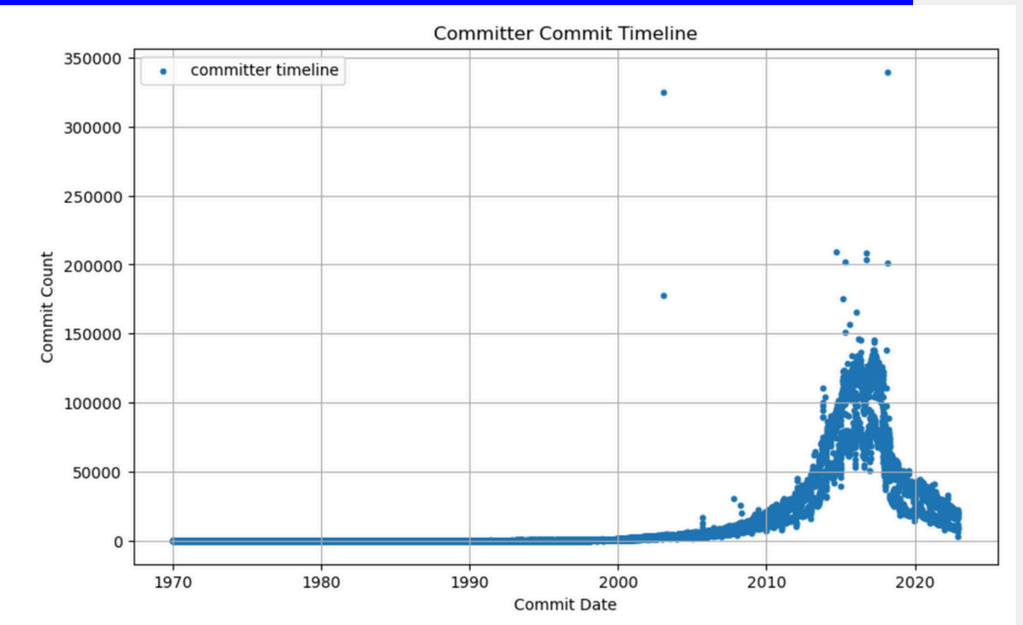
EDA - Timezone Distribution

I looked into the time offset variable within the commits table amongst the top 100 repositories (with the most commit numbers). Suprisingly, the time offset variable doesn't seem to be defined in a unified way, with a lot of values blow -12 hours and above 12 hours. So I filtered out the ones that don't make any sense and here is the real distribution of timezones.

Not surprisingly, most frequent commits were made in timezone -7, which is the US west coast and midwest.







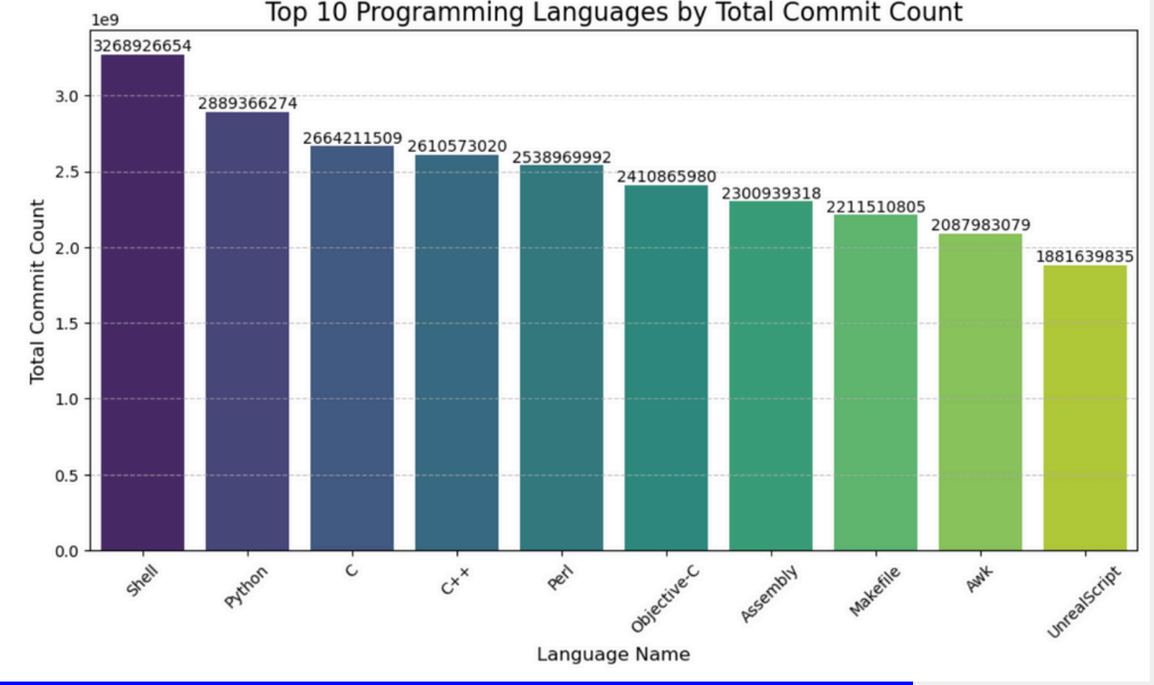
03

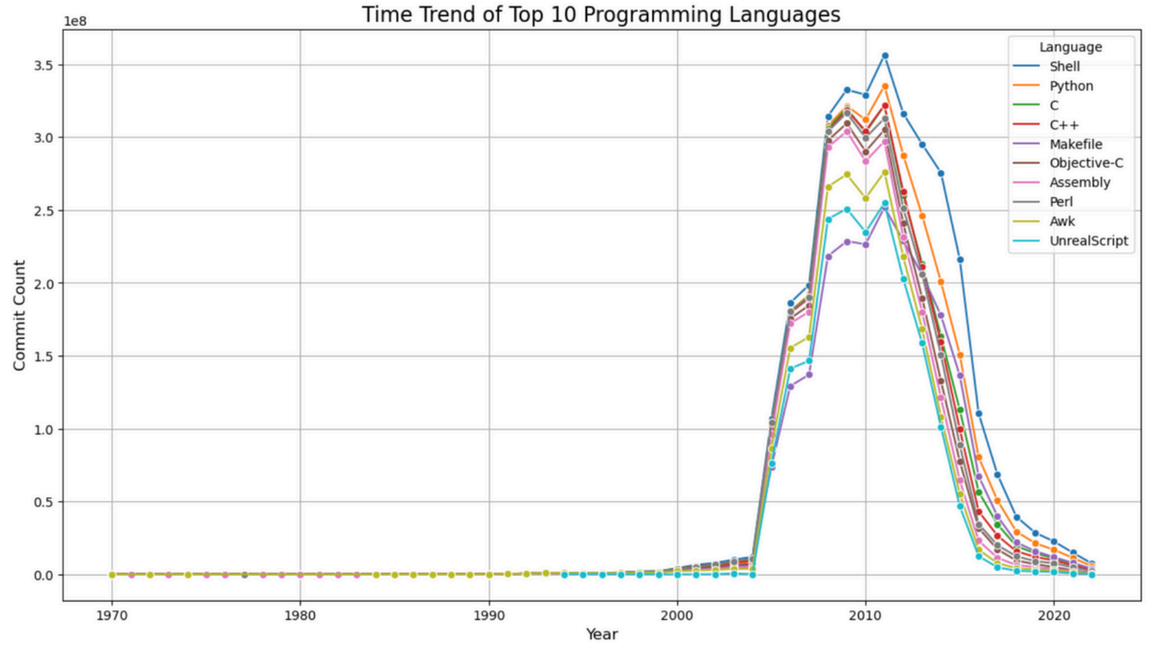
Timeline of Authorship Commits

Top graph - This scatterplot shows the count of author creation vs time,

Middle graph - This scatterplot the count of commit vs time.

Both show an increasing trend starting in early 2000s up to 2017, and then a sudden drop in count between 2017 until now.



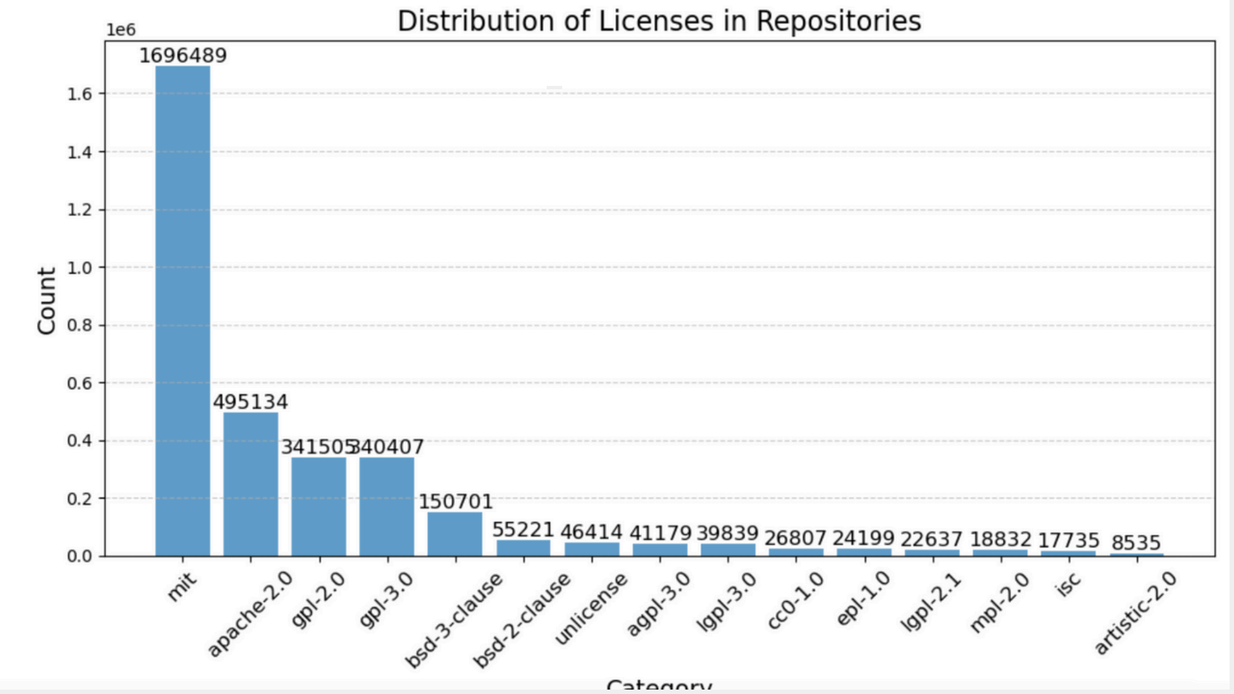


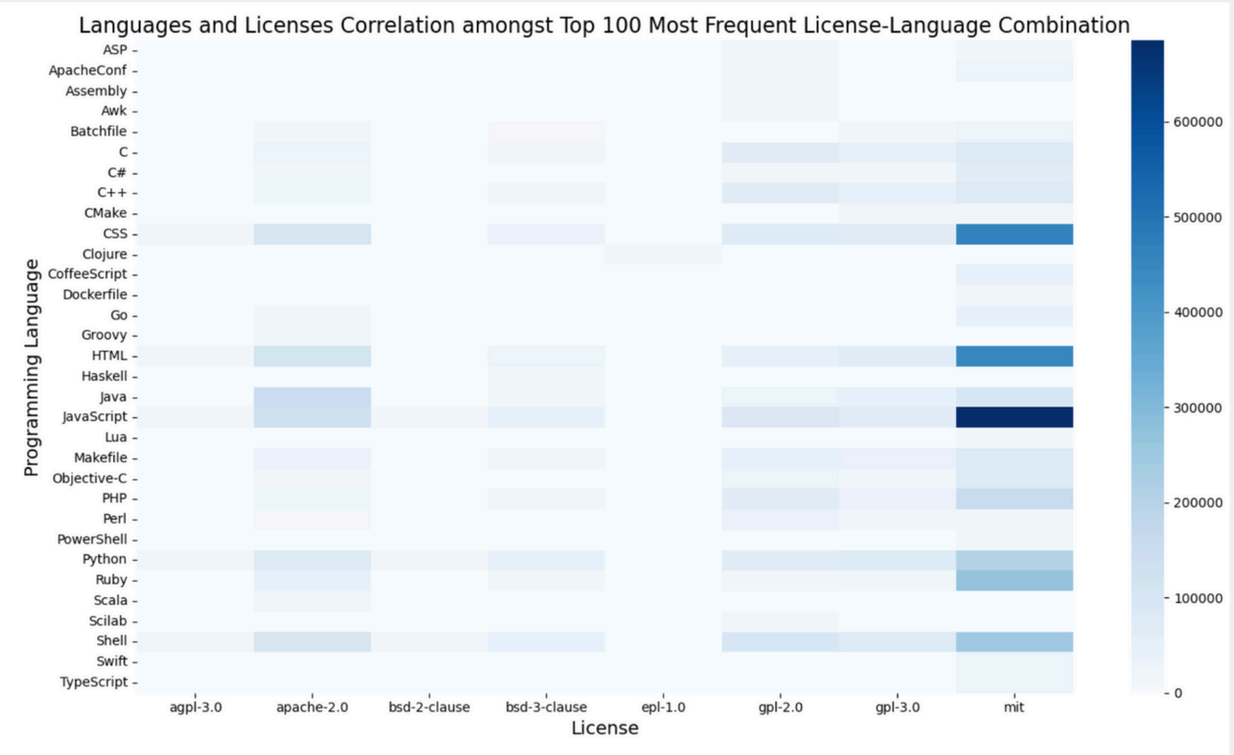
03 Timeline of Programming Languages

Top graph - This bar chart shows the top 10 most popular programming languages, calculated by total historical commit counts. We see the top one being Shell, then Python, C, C++, Perl, Objective C, Assembly, Makefile, Awk, UnrealScript.

Bottom graph shows the trend of commit counts of these top 10 languages as time goes by. All the languages seem to have peak commit count around 2012, and follow the same trend of decreasing since then. Makefile was not always that popular. Makefile became more popular than most languages after 2013.

04 Distribution of Licenses & Languages

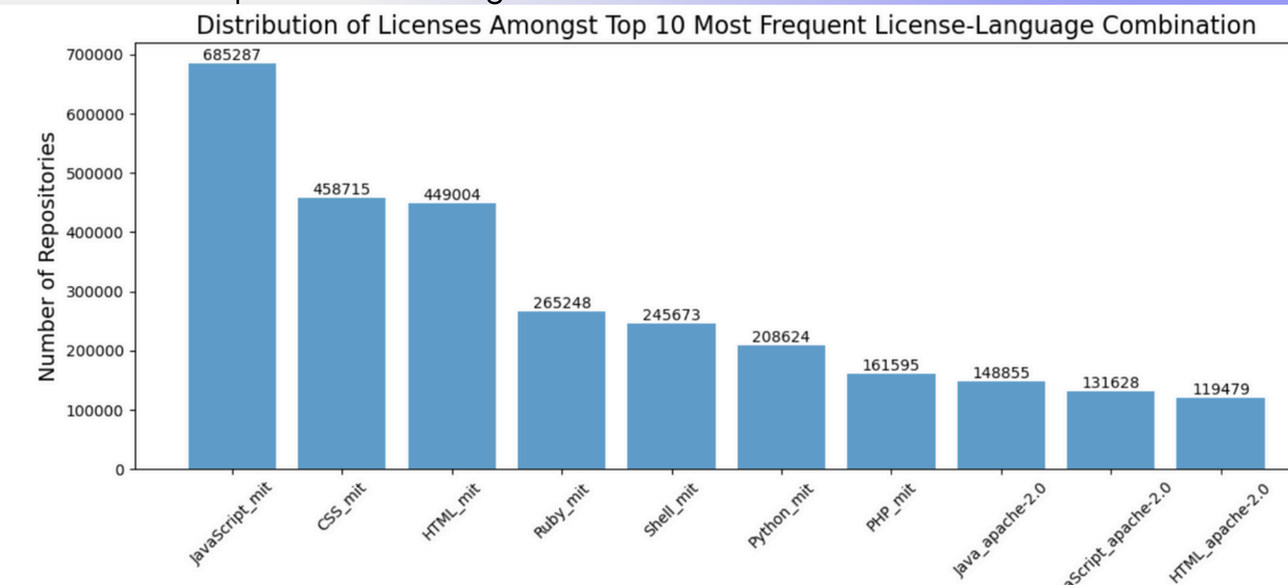




Top left graph - The most used language-license combination is JavaScript-MIT with 685 thousand total repositories using this pair.

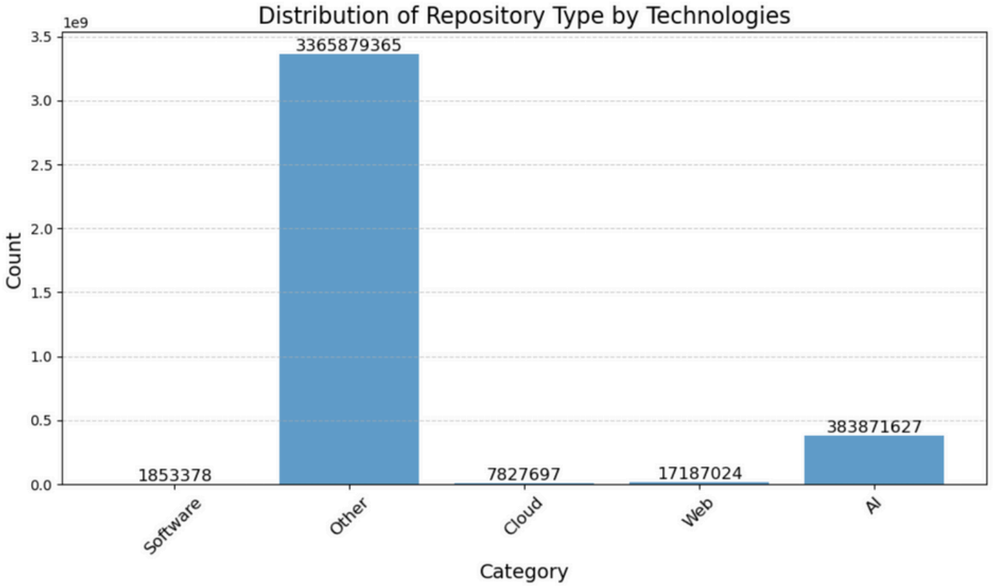
Bottom left graph - The heatmap shows high correlation between MIT and most languages, with JavaScript, HTML, and CSS being the top correlated language.

Bottom right graph - The most used license is MIT license, with 1.6 million repositories using it.

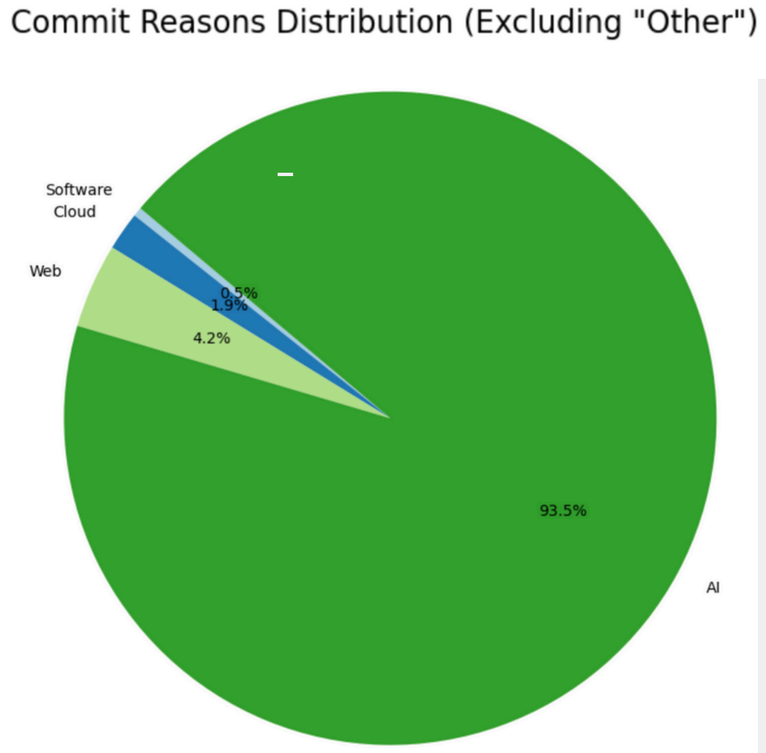


05

repo_name_exploded	commit_count
chromium/chromium	1197167
shenzhouzd/update	1188925
scheib/chromium	1104843
cminyard/linux-live-app-coredump	1085901
frustreated/linux	1074488
fabiocannizzo/linux	1071354
mpe/powerpc	1057080
rperier/linux	1053624
tprrt/linux-stable	1044596
HinTak/linux	1014842



Popular Repositories



Top left graph - This table shows the top 10 repositories with the most frequent commits. Chromium is the repository owned by Google. We also see a lot of linux-related repositories on the top ten chart.

Top and bottom right grapsh - The bar chart shows the number of repositories categorized by the type of technologies they used. I categorized them by identifying keywords in the commit subjects. Although there exists a lot of other technologies that are not categorized here, AI-related repos seem to receive the most commits, with 93% of all the commits (excluding uncategorized repos).

34.3%

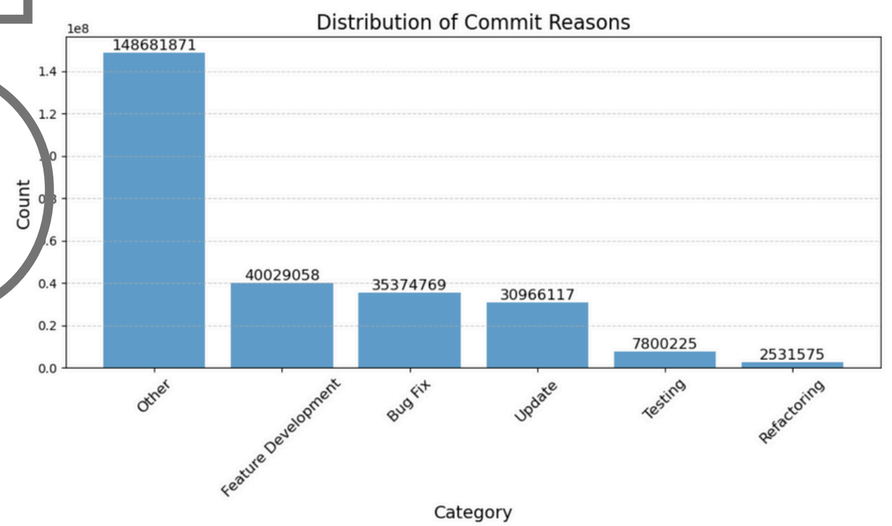
of commits to repos were related to feature development

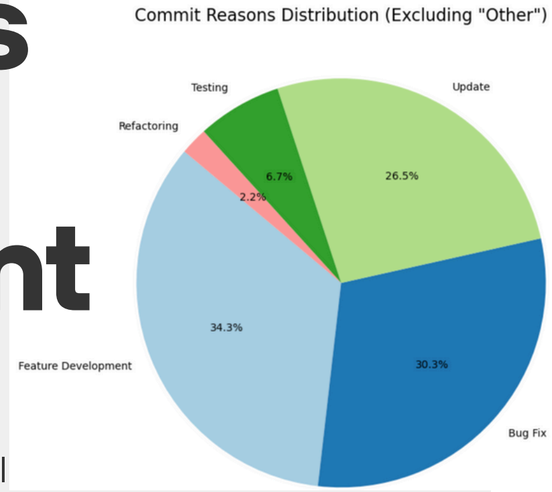
05

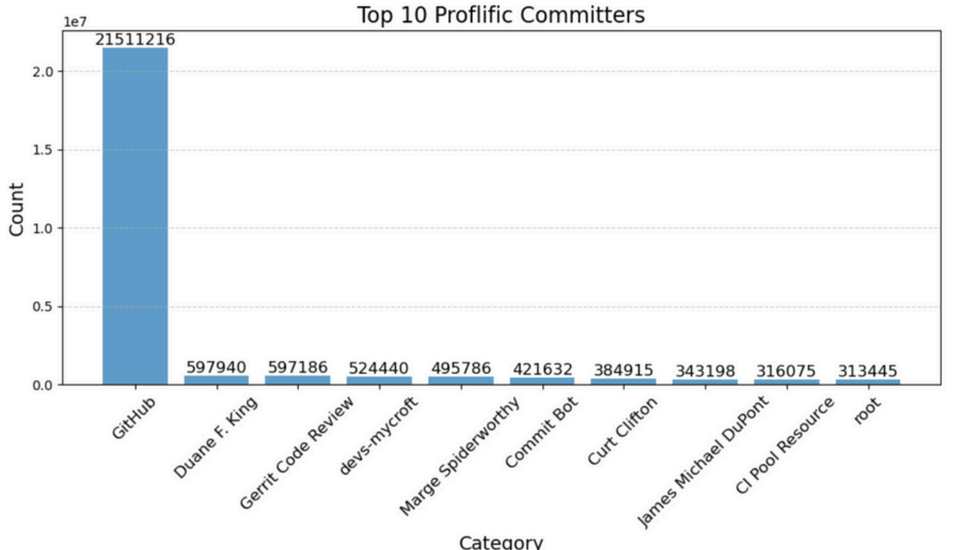
The distribution of commit reasons are shown here.

Top graph & middle graph - The histograms shows all commit reason categories, each calculates the count of commit messages attributed to the 6 categories of commit reasons. Besides the unclear type “Other”, the top commit reason was “Feature Development”, which had a total of 40 million commit messages, which was 34.6% of all commits.

Bottom graph - The bar chart shows the top 10 most prolific commuters. The top one is by Github itself with 21.5 million commits. The rest of the 9 all committed around the hundreds of thousands total commits.

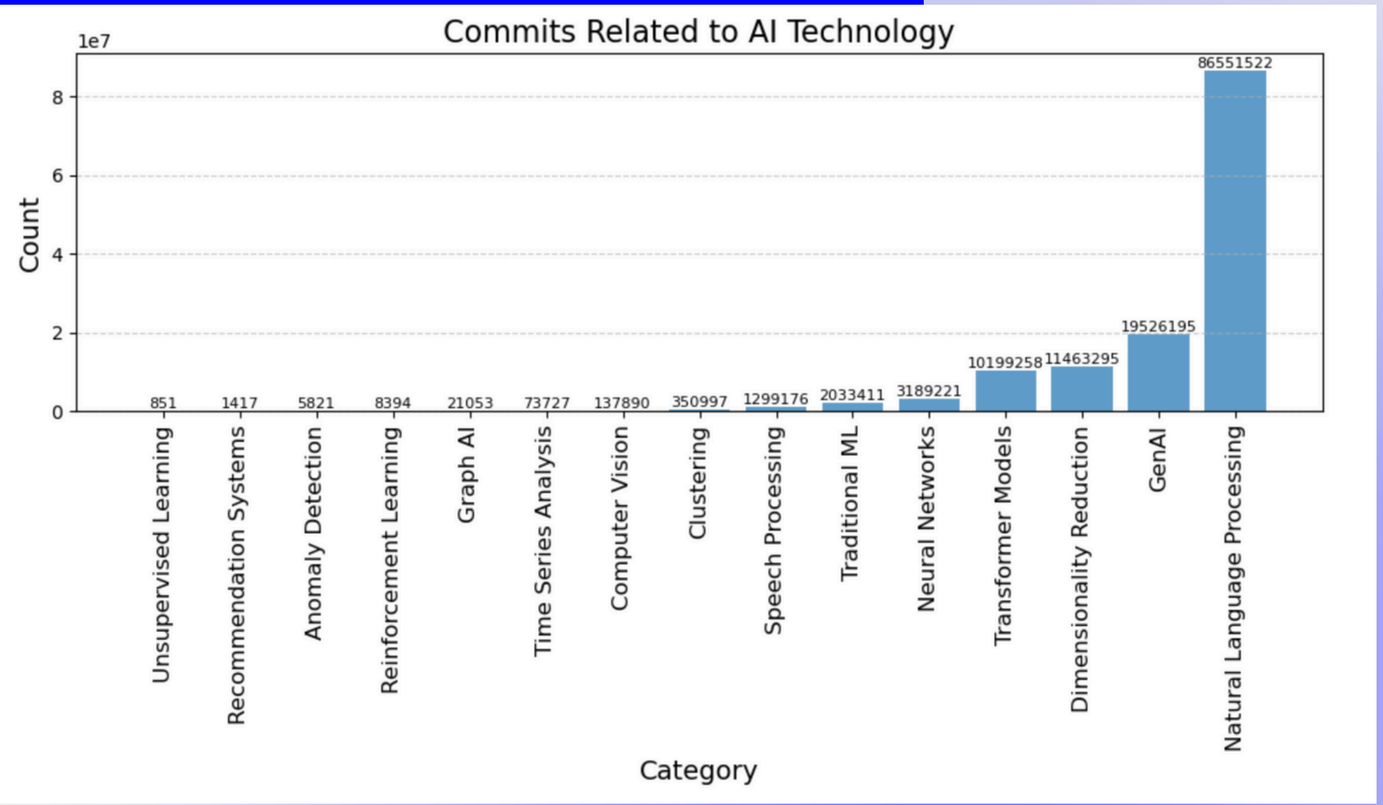






05

Most Popular AI Technology Based On Commit Counts



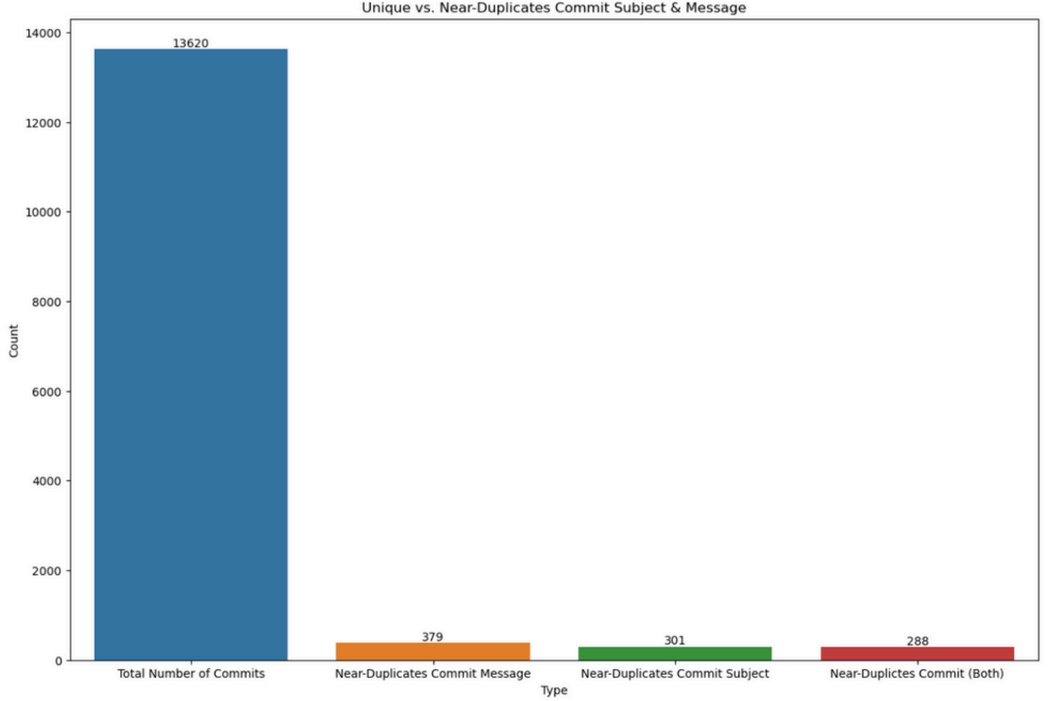
The histogram on the left shows the number of commits in each AI-related technology based on commit subjects.

Surprisingly, the top AI technology is NLP (natural language processing), with over 86 million commits having subject title related to NLP. Then follows GenAI with over 19 million commits. Then Dimension Reduction, and transformer models.

The least popular AI-tech commit subjects are unsupervised learning, and recommendation systems.

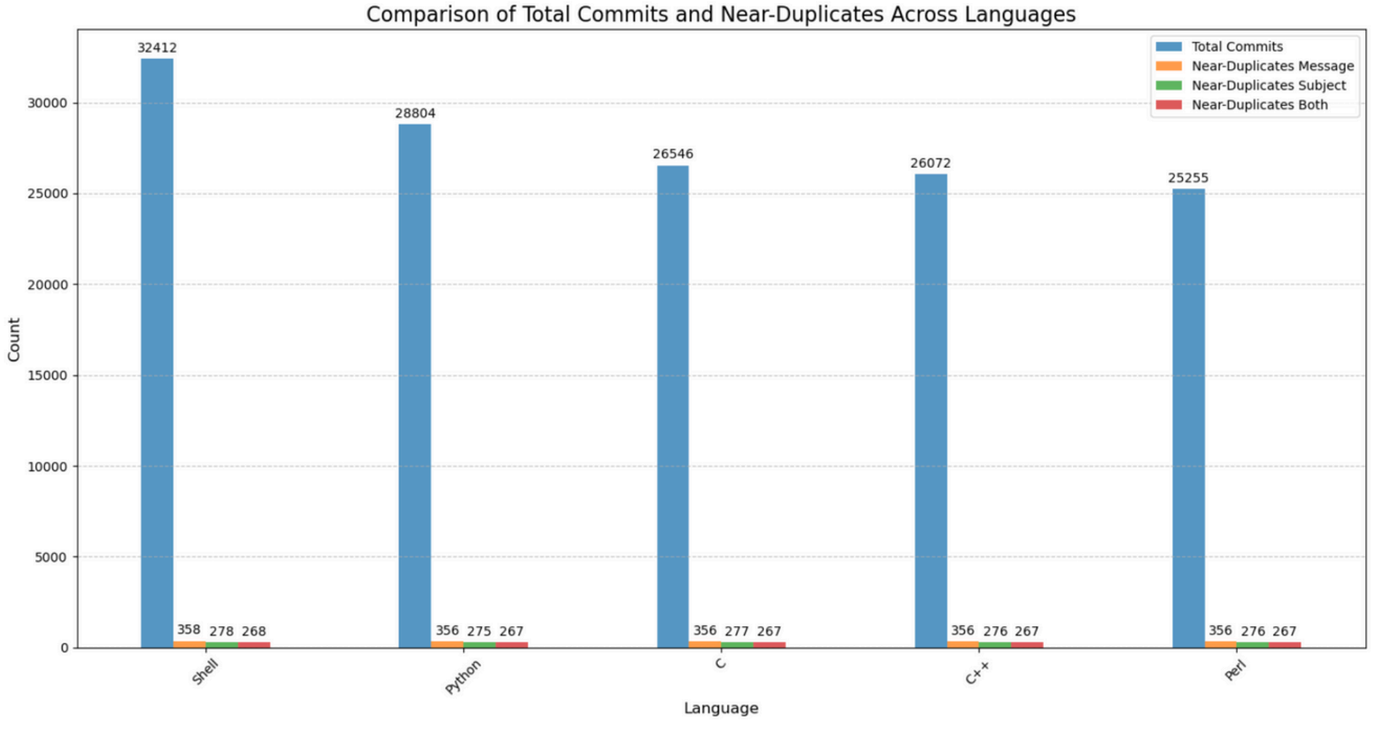
Please see Appendix 1 for how I categorized each AI technologies with keyword searching.

06 Commit Subject & Message Similarity



Top graph - I randomly sampled 0.0001 of the original commit data, and compared the text similarity amongst these commits' subject and messages. The bar chart shows that out of all 13620 unique commits, 379 pairs of commit messages are the similar, 301 pairs of commit subjects ae similar, and 288 pairs of commits have both messages and subjects similar.

Bottom graph - Then I ran text similarity comparison from the same sample, grouped by different programming languages used, and found that the message and subject similarity count was similar.



Summary and Recommendations

Conclusions:

1. Stagnation in New Activity: A notable decline in commits since 2017 suggests saturation or shifts in focus among developers.
2. AI Technology Leadership: NLP stands out as the most actively committed AI technology, highlighting its critical role in contemporary tech.
3. Importance of Licensing: The dominance of MIT licenses suggests a developer preference for permissive licensing models.
4. Makefile Emergence: The rise of Makefile indicates an increasing need for build automation tools in software development.

Recommendations:

1. Promote Emerging Languages and Frameworks: Invest in resources and tooling for less prominent languages like Makefile, which are gaining traction.
2. Capitalize on AI Growth Areas: Focus on NLP and Generative AI to align with high-growth, high-commitment trends.
3. Encourage Permissive Licensing: Advocate for and educate on the benefits of MIT licenses to align with industry preferences.
4. Address Declining Trends: Investigate underlying causes of the post-2017 decline in commits to adapt strategies for maintaining repository engagement.

Appendix 1

Keyword search for AI technology

```
from pyspark.sql.functions import when, lower, col

commits_spDf = commits_spDf.withColumn(
    "AI_tech",
    when(
        lower(col("subject")).rlike("generative|genai|gpt|chatgpt|rag|retrieval augmented generation"),
        "GenAI"
    ).when(
        lower(col("subject")).rlike("transformer|bert|roberta|distilbert|xlm|t5|llm|large language model"),
        "Transformer Models"
    ).when(
        lower(col("subject")).rlike("neural network|cnn|convolutional neural network|rnn|recurrent neural network|lstm|gru|deep learning"),
        "Neural Networks"
    ).when(
        lower(col("subject")).rlike("computer vision|image recognition|object detection|image segmentation|opencv"),
        "Computer Vision"
    ).when(
        lower(col("subject")).rlike("natural language processing|nlp|text analysis|text mining|ner|named entity recognition|text summarization"),
        "Natural Language Processing"
    ).when(
        lower(col("subject")).rlike("reinforcement learning|q-learning|policy gradient|actor-critic|dqn|deep q network"),
        "Reinforcement Learning"
    ).when(
        lower(col("subject")).rlike("machine learning|random forest|decision tree|support vector machine|svm|logistic regression"),
        "Traditional ML"
    ).when(
        lower(col("subject")).rlike("speech recognition|speech synthesis|text-to-speech|tts|automatic speech recognition|asr"),
        "Speech Processing"
    ).when(
        lower(col("subject")).rlike("anomaly detection|outlier detection|fraud detection"),
        "Anomaly Detection"
    ).when(
        lower(col("subject")).rlike("recommendation system|collaborative filtering|content-based filtering"),
        "Recommendation Systems"
    ).when(
        lower(col("subject")).rlike("clustering|k-means|hierarchical clustering|dbscan"),
        "Clustering"
    ).when(
        lower(col("subject")).rlike("dimensionality reduction|pca|principal component analysis|tsne|umap"),
        "Dimensionality Reduction"
    ).when(
        lower(col("subject")).rlike("time series|forecasting|arima|lstm for time series|prophet"),
        "Time Series Analysis"
    ).when(
        lower(col("subject")).rlike("graph neural network|gnn|graph embedding|node2vec|graph attention network"),
        "Graph AI"
    ).when(
        lower(col("subject")).rlike("unsupervised learning|self-supervised learning|contrastive learning"),
        "Unsupervised Learning"
    ).otherwise("Other")
)
```