Report on Analysing and Tracking the Trends of Coding Languages on Social Media

By,

Cheng Yuan S3832876

Sajeevan Selvaraja S3975965

Ullas Vijaya Bhanu S3872652

Table of Contents

[Introduction 3](#_Toc148703846)

[Objective 4](#_Toc148703847)

[Data Collection 5](#_Toc148703848)

[Data Preprocessing 6](#_Toc148703849)

[Data Exploration 8](#_Toc148703850)

[Sentiment Analysis Approach 9](#_Toc148703851)

[Sentiment Analysis 11](#_Toc148703852)

[Topic Modelling 14](#_Toc148703853)

[Community Detection 14](#_Toc148703854)

[Conclusion 14](#_Toc148703855)

[References 14](#_Toc148703856)

# Introduction

In the age of digital interconnectedness, social media stands as a virtual arena where ideas flow, opinions emerge, and communities coalesce. The coding landscape, a dynamic realm shaped by innovation and collaboration, finds a significant presence on various social platforms. To unravel the layers of this dynamic ecosystem and understand the heartbeat of coding language preferences, our approach extends beyond mere observation. By harnessing the power of social media data analysis, encompassing sentiment modelling, topic modelling, and network modelling, we embark on a journey to dissect the nuances that define the popularity of coding languages.

Social media platforms serve as vast repositories of user-generated content, reflecting the real-time pulse of discussions among developers, enthusiasts, and experts. This data, rich in its diversity, encapsulates not only the quantitative metrics of popularity but also the qualitative aspects of sentiments, perceptions, and emergent trends. Through sentiment modelling, we aim to decode the emotional tone surrounding coding languages, discerning whether the discourse is marked by enthusiasm, scepticism, or a neutral stance.

The language used in social media discussions is a goldmine of insights. Textual analysis allows us to delve into the semantics, syntax, and semantics of the conversations. By employing social media analytics techniques, we can unravel the implicit meanings, identify key phrases, and discern linguistic patterns that underpin the narratives around coding languages. This, in turn, contributes to a nuanced understanding of the factors influencing language adoption.

Beyond individual sentiments, social media is a crucible where topics and trends crystallize. Through topic modelling, we aim to identify clusters of discussions, uncovering prevailing themes and subjects that resonate within the coding community. This not only sheds light on current preferences but also provides a predictive lens into the potential trajectories of coding language evolution.

In essence, our exploration goes beyond the surface metrics of popularity. We delve into the socio-linguistic fabric of social media, using analytical tools to unravel the subtleties that shape the ebb and flow of coding language preferences. Through sentiment modelling, topic modelling and network modelling, our endeavour is to decode the unspoken narratives within the digital corridors where coding languages thrive.

# Objective

The primary objective of this comprehensive study is to conduct an in-depth analysis of social media data to unravel the multifaceted dynamics influencing the popularity of coding languages. Our approach integrates analytical methodologies, including sentiment modelling, topic modelling, and network analysis, to provide a nuanced understanding of the underlying factors shaping the coding landscape. Through this exploration, we aim to achieve the following key objectives:

1. The first objective revolves around employing sentiment modelling as a lens to understand the emotional context enveloping discussions on coding languages. Through sentiment analysis, we seek to discern not just the popularity of languages but also the underlying emotional tones that influence developer sentiments. By identifying trends of positivity, negativity, or neutrality, we aim to uncover the visceral responses that contribute to the adoption or avoidance of specific coding languages. This objective allows us to capture the essence of the human experience within the coding community and understand the emotional drivers behind language preferences.
2. The second objective focuses on topic modelling as a means to identify clusters of discussions and prevailing themes within the coding community. Through topic modelling algorithms, we aim to uncover latent topics that dominate social media conversations related to coding languages. This goes beyond individual sentiments and provides a thematic landscape, allowing us to understand the broader context in which coding languages are discussed. By identifying emerging topics, we can offer insights into current preferences and anticipate potential shifts in the programming landscape.
3. The third objective extends our analysis to the collaborative ecosystems within coding communities. Network analysis involves mapping out relationships, collaborations, and influential nodes within the social network of developers. By understanding the dynamics of connections, community structures, and the role of influencers, we gain insights into how coding languages proliferate within these collaborative networks. This objective enables us to go beyond individual preferences and explore the social dynamics that contribute to the diffusion of coding languages.
4. The final objective involves synthesizing findings from sentiment analysis, textual analysis, topic modelling, and network modelling to achieve a holistic understanding of the factors influencing the popularity of coding languages. By integrating quantitative and qualitative insights, we aim to provide a comprehensive narrative that goes beyond surface-level metrics. This holistic approach allows us to unravel the intricate interplay of emotional responses, linguistic patterns, thematic landscapes, and collaborative networks, offering a nuanced perspective on the complex dynamics shaping coding language preferences.

These objectives can further be tweaked in order to contribute to the following applications based on the use case:

* Provide developers, industry stakeholders, and coding language enthusiasts with actionable insights derived.
* Understand the collaborative dynamics within coding communities.
* Contribute valuable empirical data and methodologies to the broader field of social media and network analysis research.
* Foster a deeper understanding of the socio-technical factors that influence the popularity of coding languages, bridging the gap between quantitative metrics and the rich tapestry of human interactions within the programming community.
* Offer predictive insights into the future trajectories of coding languages.

# Data Collection

The data collection stage is like laying the cornerstone of a grand structure in social media and network analysis. It's the initial and crucial step that sets the tone for the entire investigative process. In social media and network analysis, it's not just about amassing raw information; it's about carefully curating the right pieces to form a mosaic of insights. Social media, being a dynamic and ever-evolving space, is a goldmine of user-generated content, conversations, and relationships. At this early stage, we cast our net wide to capture the rich tapestry of interactions, sentiments, and collaborations that unfold across platforms.

The significance of this step lies in its power to shape the narrative we're about to uncover. It's not just about quantity but also about quality—selecting data points that encapsulate the diversity of voices, opinions, and trends within the coding community. This careful curation ensures that our subsequent analyses reflect the true pulse of the digital landscape, providing a nuanced understanding of how coding languages ebb and flow in popularity.

Moreover, data collection is not a one-size-fits-all endeavour. It involves making strategic choices about the platforms from which we gather information. In our case, opting for Reddit and Stack Overflow adds layers to our exploration. Reddit, with its decentralized structure and diverse subreddits, becomes a canvas for understanding community sentiments. On the other hand, Stack Overflow, with its structured Q&A format, offers a lens into the practical aspects of language usage, like addressing programming challenges and gauging language popularity over time.

In essence, this pivotal step is not just about acquiring data; it's about laying the foundation for a journey of discovery. The data we collect is not static; it's a living, breathing entity that encapsulates the myriad voices and narratives within the coding ecosystem. And as we embark on the subsequent stages of analysis, it is this foundation that will shape the depth and richness of the insights we uncover, ultimately contributing to a more profound understanding of the dynamics of coding languages in the digital realm.

We've chosen two platforms—Reddit and Stack Overflow—to get a well-rounded view of the coding community. Reddit gives us insights into sentiments and community trends due to its diverse discussions. Stack Overflow, on the other hand, offers a structured space for questions and answers, letting us understand trends and the popularity of languages over time. Some of the reasons behind choosing these two platforms for our analysis are listed below –

* Reddit and Stack Overflow cater to different aspects of the coding community. Reddit's subreddits provide a range of sentiments and community-related trends. Stack Overflow's Q&A format, focused on programming challenges, helps us see the popularity of languages in a more structured way. This combo ensures we cover both qualitative and quantitative aspects.
* Reddit is a place for open discussions, making it great for understanding sentiments and community dynamics. Different subreddits show us different sentiments, helping us understand how emotions play a role in coding language discussions.
* Stack Overflow's Q&A format gives us a structured dataset on programming challenges. By looking at questions related to programming languages, we can see trends in language popularity over time. It's a practical way to understand what languages developers turn to for solving real coding issues.
* Both platforms allow us to track trends over time. Stack Overflow's historical data lets us see how language preferences have changed, and Reddit gives us a real-time view of community sentiments. This mix helps us understand the evolution of preferences and sentiments.
* Both Reddit and Stack Overflow naturally capture community interaction. Using network analysis, we can map out relationships and influential users, giving us insights into how coding communities collaborate. It's a way to see the social side of language adoption.

# Data Preprocessing

In the world of social media and network analysis, data pre-processing acts as the meticulous curator, shaping raw information into a refined masterpiece. This crucial phase involves a sequence of tasks that cleanse, organize, and transform the collected data into a state ready for in-depth analysis. When we collect data from social media platforms, it often arrives in a raw and unstructured state, laden with noise, inconsistencies, and irrelevant elements.

As the information of social media is user-generated, content varies widely in language, format, and expression and therefore data-preprocessing becomes an integral step before performing any analysis on the collected data. It ensures that the language used in conversations is standardized, making it easier to discern the emotional tone. Duplicates, irrelevant information, and variations in expression are sifted through, allowing sentiment analysis to focus on the genuine emotional pulse within the data.

The true significance of data pre-processing extends to its role in enhancing the quality and reliability of analyses, especially in sentiment analysis and topic modelling. By cleaning up the messy room of data, it streamlines the analytical journey. Now, researchers can delve into understanding emotional nuances, thematic landscapes, and collaborative networks without being hindered by unprocessed information.

In the process of preparing the Reddit data collected from the 'r/programming' subreddit, we've leveraged the PRAW library in Python for gathering the top posts and comments, we recognize the importance of refining the textual content for meaningful analysis. To achieve this, we've employed a set of tools, specifically the TweetTokenizer from the NLTK library, in conjunction with a list of English stopwords and punctuation.

The use of tweetTokeniser from NLTK allows us to go beyond conventional tokenization, catering to the unique linguistic patterns often found in social media content, including Twitter-like and Reddit-like phrase. This specialized tokenizer helps in breaking down the textual data into meaningful units, capturing the essence of discussions in the 'r/programming' subreddit. Additionally, we've compiled a list of English stopwords, common punctuation, and the addition of 'via' to filter out noise and focus on the core content.

To further streamline the textual data, we've incorporated the Porter Stemmer from NLTK. Stemming involves reducing words to their root or base form, helping to unify variations of the same term. The tweetStemmer allows us to achieve this, enhancing the consistency of our term frequency counter.

With these pre-processing tools in place, we can now generate a term frequency counter. This counter provides insights into the frequency of each term within the pre-processed textual data. It is a fundamental step in understanding the significance and prevalence of specific words or phrases in the discussions within the 'r/programming' subreddit.

This meticulous pre-processing approach not only ensures that the data is cleaned from noise but also facilitates the generation of meaningful insights through the term frequency counter. The choice of specialized tokenization and stemming techniques tailored for social media content aims to capture the essence of discussions in the programming community, setting the stage for a more refined and insightful analysis of the collected Reddit data.

In the context of processing Stack Overflow data, our approach is notably streamlined due to the inherent structure of questions and the availability of tags. Since questions on Stack Overflow are tagged with relevant programming languages, we find ourselves in the advantageous position of not requiring extensive pre-processing. The richness of data lies in the specificity of tags, each serving as a categorical marker for the programming language associated with a given question.

Our strategy, exemplified in the code snippet retrieving the top trending languages, revolves around leveraging the Stack Exchange API to fetch tag data directly. This means that the data we receive is inherently focused on programming languages, as tags are explicitly linked to these languages in the Stack Overflow ecosystem. Therefore, our pre-processing steps primarily entail filtering out the noise and focusing on what matters—programming languages. The script intelligently targets specific languages of interest, such as Python, JavaScript, Java, and others, ensuring that the resultant data is a curated collection of tags relevant to our analysis.

In essence, the beauty of Stack Overflow lies in its structured approach to content organization. The meticulous tagging system, where each question is associated with one or more programming languages, obviates the need for extensive pre-processing. Instead, our focus shifts to refining the data by extracting only the tags that correspond to programming languages. This strategic filtering sets the stage for a more direct and meaningful exploration of language-specific trends and discussions within the Stack Overflow community.

# Data Exploration

# Sentiment Analysis Approach

Sentiment Analysis, also known as opinion mining, stands as a formidable computational method aimed at unravelling sentiments, opinions, and emotions embedded within textual data. This technique serves the crucial objective of discerning the subjective tone behind a piece of text, categorizing it into positive, negative, or neutral sentiments. Operating within the domain of Natural Language Processing (NLP), Sentiment Analysis harnesses a blend of machine learning and linguistic methodologies to comprehend the emotional nuances conveyed through written or spoken language.

In its essence, Sentiment Analysis acts as a linguistic detective, peeling back the layers of text to uncover the underlying sentiments that colour the expressions of individuals. It's not merely about deciphering words but understanding the emotional landscape encapsulated within them. This analysis provides a lens through which we can fathom the attitudes, perspectives, and reactions of individuals or communities, offering valuable insights into the subjective aspects of communication.

In the context of programming languages, Sentiment Analysis emerges as a potent tool for decoding the sentiments within discussions on platforms like Reddit. We have thus performed Sentiment Analysis on two such popular subreddit pages, namely – ‘r/programming’ and ‘r/programmingLanguages’ in order to achieve insights about the below objectives –

* By applying Sentiment Analysis to posts and comments in the above-mentioned programming-related subreddits, we can gauge the sentiments expressed toward various programming languages. Positive sentiments might indicate enthusiasm, successful experiences, or appreciation for a language, while negative sentiments could signal challenges, frustrations, or criticisms.
* Additionally, they also shed light on the general consensus of the developer community towards programming techniques, good practices and newly released software tools and platforms.
* Sentiment Analysis also allows us to track the adoption of programming languages over time. An upsurge in positive sentiments might correlate with the growing popularity of a language, showcasing excitement and interest within the community. Conversely, a spike in negative sentiments could indicate issues or concerns affecting the perception of a language.
* The analysis can also help uncover the experiences of developers with different languages. Positive sentiments may signify satisfaction with a language's features or community support, while negative sentiments could highlight difficulties, bugs, or dissatisfaction. This nuanced understanding is invaluable for developers and language creators alike.
* Sentiment Analysis acts as a radar for emerging trends within the programming community. By monitoring sentiments, we can identify languages gaining traction, anticipate shifts in preferences, and stay attuned to the ever-evolving dynamics of the programming landscape.

Now that we have meticulously pre-processed and explored our collected data, it's time to dive into the implementation approach of our Sentiment Analysis. In navigating the diverse approaches available, we've chosen the VADER (Valence Aware Dictionary and Sentiment Reasoner) method to measure and interpret the emotional tones encapsulated in our textual discussion. For our analysis, we have used the Natural Language Toolkit’s (NLTK) ‘vader’ package for this functionality.

VADER operates on a sophisticated sentiment lexicon, a meticulously curated dictionary that assigns sentiment scores to words based on their emotional connotations. This approach goes beyond a simplistic word counting mechanism; it delves into the intricacies of language by considering not just the presence of sentiment-bearing terms but also their intensity and contextual nuances.

What sets VADER apart is its ability to capture the intricate interplay of language, even when confronted with the challenges of informal expressions, slang, and complex linguistic structures. It achieves this by computing four distinct sentiment scores for each token within the text. These scores include positive, negative, and neutral sentiments, each quantifying the respective emotional intensity. However, the crux lies in the compound score, a synthesized metric that distils the overall sentiment polarity of the text into a single value.

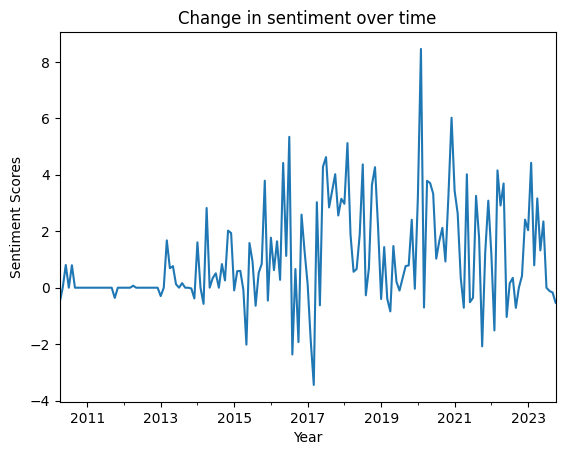
The computation process involves a range from -1 to +1, where a score of -1 signifies highly negative sentiment, +1 indicates highly positive sentiment, and 0 represents a state of neutrality. This nuanced approach allows us to capture the shades of sentiment within our textual data, offering a comprehensive understanding of the emotional tones embedded in discussions related to programming languages.

Programming languages discussions often involve technical jargon and domain-specific terms. VADER's lexicon is finely tuned to comprehend the sentiment behind such language, allowing it to capture the emotional nuances even in the presence of highly specialized and technical vocabulary. The SentimentIntensityAnalyzer function from the NLTK package not only identifies sentiment polarity but also measures the intensity of each sentiment—positive, negative, and neutral. In programming discussions, where sentiments can range from enthusiastic support to frustrated criticisms, this intensity measurement thus provides a more nuanced understanding of the emotional spectrum.

Additionally, in order to enhance the accuracy of our SentimentIntensityAnalyzer function, we have used the TweetTokenizer() object from the NLTK package to tokenize the posts and comments data. The TweetTokenizer is not just a generic text tokenizer; it's a specialized tool tailored for the idiosyncrasies of social media language. In the context of analyzing sentiments and discussions related to programming languages on platforms like Reddit, this tokenizer ensures that the unique aspects of online communication are preserved and appropriately represented during the Sentiment Analysis process.

# Sentiment Analysis

The observed results in sentiment fluctuations within the 'r/programming' and 'r/ProgrammingLanguages' subreddits, plotted against the years, provide interesting insights into the evolving emotional landscape of these online communities. These can be seen in Figures 1 and 2 below –

  
Figure 1: Change in sentiment over the years in the subreddit ‘r/programming’.

From Figure 1, we find that during the initial years from 2011 to approximately mid-2015, the sentiment scores exhibit a remarkable stability, consistently hovering close to the neutral line. This can be attributed to the subreddit space being in its initial stages and thus not being very active.

A significant shift occurs post mid-2015, marked by more pronounced fluctuations in sentiment. This inflection point indicates an increase in activity on the ‘r/programming’ subreddit page and also can indicate a change in the emotional landscape of the involved group of people. The community's collective sentiment becomes more dynamic, responding to evolving factors that trigger varied emotional responses.

The standout feature is the prominent peak around 2019, where the sentiment score reaches an exceptionally high positive value, exceeding a significant score of 8. This peak signifies an exceptionally positive sentiment during that particular period. The community likely experienced a surge of enthusiasm, positive discussions, or shared successes, contributing to this remarkable high.

Conversely, there are notable troughs in sentiment, particularly around 2017 and 2022, where sentiment scores dip below 0, indicating negative sentiment. These troughs may align with specific events, controversies, or challenges within the programming community that triggered a more negative or critical response.

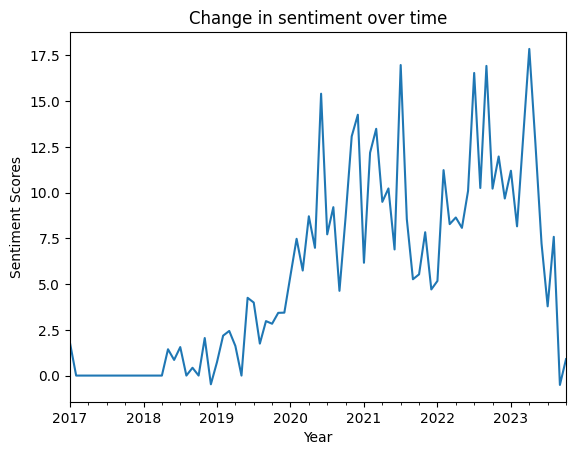
In the more recent years spanning 2021 to 2023, the sentiment graph depicts a generally oscillating pattern. During this period, the community experiences a mix of positive and negative sentiments. The oscillations suggest a dynamic and diverse range of emotions, indicating that discussions within the subreddit are marked by both optimistic and critical viewpoints.

The overall graph encapsulates the dynamic nature of sentiment within the 'r/programming' subreddit over the years. While certain periods exhibit stable sentiments, there are distinct phases marked by either heightened positivity or notable negativity. These fluctuations are indicative of a community that actively responds to changing dynamics in the programming landscape, reflecting the diverse perspectives and emotional nuances inherent in discussions around programming languages and related topics.

Similarly, looking at the change in sentiment over the years in the ‘r/programmingLanguages’ subreddit shown in Figure 2, we find that there seems to be a discernible increase in sentiment from 2017 to early 2020. During this period, the community seems to experience a positive shift in sentiment, suggesting a growing sense of optimism, collaboration, or shared positive experiences related to programming languages.

Post-2020, the sentiment becomes more volatile. The stability observed in the earlier years gives way to fluctuations, indicating a potentially more dynamic and diverse emotional landscape within the 'r/ProgrammingLanguages' subreddit. This can be attributed most of the global developing community moving to remote work and changing their methods of working.

Despite the increased volatility post-2020, there is a general upward trajectory in sentiment until 2022. This suggests that, on the whole, the community maintains a positive sentiment trend during this period. The programming language discussions within the subreddit seem to be characterized by an overall optimistic and constructive atmosphere.

  
Figure 2: Change in sentiment over the years in the subreddit ‘r/programmingLanguages’

However, towards late 2022 and into 2023, a notable decline in sentiment is observed. The sentiment scores sharply decrease, bringing them down to around 5 by the end of 2023. This decline may indicate a shift in community dynamics, possibly influenced by specific events, changes in discourse, or external factors impacting the programming language discussions.

The sentiment experiences a sharp decline in late 2022 to 2023, bringing the scores back down to around 5 by the end of 2023. This decline suggests a shift in community sentiment dynamics, potentially influenced by changing discussions, external events, or evolving community interactions.

While the community witnesses an overall upward trajectory in sentiment until 2022, the subsequent decline indicates a fluctuating and evolving emotional landscape. These fluctuations may be influenced by a myriad of factors, including technological shifts, community dynamics, or global events impacting the programming language discussions within the subreddit.

# Topic Modelling

# Community Detection

**Introduction:**

Implemented the Louvain method to detect communities within the social network, highlighting the clusters of conversation and common themes.

**Main Findings:**

* Prominent Coding Languages:

Python emerged as a frequently discussed coding language, suggesting its popularity in the developer community. Java, JavaScript, and C++ also had substantial mentions.

* Discourse Themes:

The data revealed several common themes in developer discourse, including general programming terms, web development, and troubleshooting coding issues.

* Community Clusters:

Distinct communities were detected, each revolving around specific coding languages or development areas. This provides insight into the pockets of discussions and knowledge-sharing occurring on the platform.

# Conclusion

# References