Managing Big Data Project Report
Stack Overflow Insights - Group 3

Tran Thi Hoang Anh
University of Twente
Enschede, The Netherlands
tranthihoanganh@student.utwente.nl

Michael
University of Twente
Enschede, The Netherlands
michael@student.utwente.nl

Quang-Hung Nguyen
University of Twente
Enschede, The Netherlands
nguyenquanghung@student.utwente.nl

Colyn Jonker
University of Twente
Enschede, The Netherlands
j.h.c.jonker@student.utwente.nl

Abstract

This paper presents a deep dive into various aspects of StackExchange's Stack Overflow data dump from 2008 to 2022. It explores programming languages and library trends, factors affecting answer quality, sentiment analysis of post contents, and applies Principle Component Analysis and t-distributed Stochastic Neighbor Embedding to the engineered post contents. The results show that while there is a strong correlation between users' reputation and the quality of answers on Stack Overflow, there is a weak link between questions' readability features and how fast they are getting answers. The study also reveals that changes in the software and technology world are also strongly reflected in tags frequency and that the majority of the posts on Stack Overflow have a positive sentiment.

Keywords: Stack Overflow, big data, sentiment analysis, feature engineering, PCA

1 Introduction

1.1 Motivation

Stack Overflow is one of the most popular question-and-answer (Q&A) sites for programmers. With thousands of questions posted and answered every day, what is behind Stack Overflow's success are their strict moderation and reputation rules: volunteer "moderators" frequently patrol threads to remove/edit flagged or unnecessary content, or only highly-reputed users can "upvote", "downvote" an answer. Questions that are duplicates or too specific are moderated accordingly. These are only two examples; there are plenty more; This makes Stack Overflow more than just a Q&A site. It can also be considered an ever-evolving library for programming, with its more than valuable data to be analyzed in different technology trends and how the focus of technology has changed since the year Stack Overflow started (2008) up until now - which is one of our two research interests in this paper. The other research motivation will focus on the quality of content on Stack Overflow, including posts and answers, which will be described in more detail in section 1.2 below.

1.2 Research questions

Two main research questions are formulated in this report:

RQ 1 : What are the popular tech topics discussed in Stack Overflow, and how has it changed over time?

RQ 2 : What are the factors contributing to the quality of questions and answers in Stack Overflow?
We break down this research questions into three aspects with the following sub-questions:

1. How do tags influence the possibility of questions getting answered?
2. How does user reputation influence the quality of Stack Overflow's posts?
3. Which content factors drive a question to be answered (and how fast)?

1.3 Outline

In the next section, existing works that are related to our research will be discussed. Section 3 will discuss different methodologies, including how the data is prepared and analyzed, among features engineering on posts content and machine learning models. The results and visualizations will be discussed in Section 4, which will be followed by Discussion and Conclusion in section 5 and 6, respectively.

2 Related work

In recent years, question-and-answer sites such as Stack Overflow have received much attention from the academic and research community, making it a popular subject for the exploration and analysis of user behavior and collaboration in online knowledge sharing. Existing studies conducted on these sites had explored an extensive area of research focus and disciplines, in particular Data Analytics, Big Data, and Machine Learning.

2.1 Literature about Stack Overflow tags

User-defined tags can be considered the starting point to measure discussion topics in Stack Overflow. [3]. There are two common approaches to analyzing Stack Overflow tags from previous studies: Tag prediction and Topic clustering. Both approaches usually combine the content of the posts and the tags to create clusters of topics rather than using only tags.

Schuster et al. (2013) have developed an automatic content tagging model using the SVM algorithm for SO questions, using a subset of documents with the 1,000 most popular tags. The author observed that more than 50% of the questions contain from two to three tags which belong to 20 programming languages[1]. However, the model has two limitations. First, predicted tags are contained in the body and or title of the post but not the real tag to the question. Second, the predicted programming languages are also not real tags because of the user's tendency to put specific tags relating to a programming language (jquery vs. javascript).[26]

For topic clustering, studies by Barua et al. (2014) [3] applied latent Dirichlet allocation (LDA), a statistical topic modeling technique, to automatically discover the main topics present in developer discussions. The corpus used in this study, however, does not include tags. Instead, the authors used posts and answers as the input for the LDA models. 40 topics are identified from this study, such as Data Management, Platform-Specific Discussions, Quality Assurance, and Collaboration Knowledge/Experience. The topics are descriptive of the issues rather than just programming languages relating to the questions. For example, the word "c#" appears in more than half of the topics. Using statistical tests for trend analysis, the authors found that mobile application development grew faster than web development, and Java was a constant player in a programming language.

This project focuses more on the tags themselves as the representation of SO questions. Clustering text at the word level has been carried out in a study by Bhakdisuparit and Fujino (2018), but with a different corpus: Twitter hashtags. The authors used the K-means algorithm with Jensen-Shannon divergence as the measure of distance for clustering English hashtags in tweets into meaningful groups.[4]. K-means clustering is probably suitable for our purpose because both SO tags and Twitter hashtags are concise and defined by users to summarize their original content. The nature of social interaction between the author of the original content and followers is another similarity between the two platforms.

[1]20 most frequent tags: C#, java, PHP, javascript, android, jquery, c++, python, iPhone, asp.net, MySQL, HTML, .net, ios, objective-c, SQL, CSS, Linux, ruby-on-rails, windows

2.2 Literature about user reputation, quality of answers and questions

User reputation or ratings has also been explored; for instance, a recent study [27] examined whether user reputation could be used as an indicator of user expertise. This study further analyzed how top users gained their reputation and concluded that user reputation is an indicator of user expertise, although it is not the only one. In Stack Overflow, aside from reputation, some badges are awarded to users based on their participation and activity in the community. According to Papoutsoglou et al. [23], these badges can be seen as a gamification method to cluster users based on their participation on the platform. Furthermore, various studies [2, 19, 28] have explored the influence of badges on user behavior. However, we found no studies correlating badges with user reputation.

The quality of questions and answers is another prominent area of research related to Stack Overflow. Several studies have analyzed the quality of answers on the platform: exploratory data analysis on answers [6, 13, 29] and assess rankings of answers using machine learning techniques [9]. Moreover, some other studies also explained and clustered the relationship between questions asked and answers received, based on a few factors, such as difficulties [12], generated values [1], and user motivation [20]. To the best of our knowledge, neither the quality of questions nor the correlation between user reputation and question-answers has been much explored by existing studies. These are what we will explore and discuss in the next sections.

2.3 Literature about feature engineering and sentiment analysis of Stack Overflow

Feature Engineering:
Hodgins, in their dissertation titled "Classifying the Quality of Questions and Answers From Stack Overflow", engineered multiple features for posts and answers such as body length, code count, spelling error count, email, URL, and spaces count, etc [14]. However, their final goal was to train a model for the classification and prediction of the quality of posts and answers. At the same time, in this project, our team focuses more on how those features correlated with response time.
Sentiment Analysis:
In general, Sentiment Analysis (SA) is a popular method for quantifying the sentiment aspect expressed by the writer towards the entity/object mentioned in the text [18]. Coupled with the rise of big data (especially large text data from social media), SA extracts a useful and prominent feature from the text which can be used for multiple purposes. There are mainly two types of Sentiment Analysis algorithms: lexical-based and machine learning based. The former used a set of pre-defined rules to classify emotion, while the latter utilized a machine learning model (either supervised or unsupervised). Novielli et al. in 2018 trained a supervised machine

Table name	Description	Raw data size
Posts	Content and metadata of posts on SO. Our focus: questions & answers	99.61 GB
Badges	Information about badges of users	5.56 GB
Users	Information about users	6.62 GB
Tags	Summary of tags in SO questions. However, this table is not linked directly to Posts.	5.7 MB

Table 1. Raw data description

learning model to specifically classify the sentiment of software developers, called Senti4SD [21] (trained on 4800 Stack Overflow questions and answers), and later compared four different SA algorithms: two lexical-based, Senti4SD and another supervised learning algorithm for the classification of software developers' sentiment [22]. Although their algorithm (Senti4SD) shows a robust performance compared to the others, we decided against implementing it due to its non-straightforward implementation and small train dataset.

Calefato et al. [7] and Hodgins [14] have also utilized a machine learning-based approach to classify the sentiment of Stack Overflow's content. While Calefato et al. argue that emotions are important when considering the successfulness of questions, Hodgins states using models and lexical rules based on social media sites' texts, such as Facebook and Twitter, might lack effectiveness for technical content on Stack Overflow.

3 Methodology

3.1 Data preparation

For this project, we use archived data from Stack Exchange Data Dump, covering the period from 2008 to 2022. The archived data about Stack Overflow only includes 8 out of 29 tables in their schema. In this project, we use four tables: Posts, Users, Badges, and Tags. Raw data can be downloaded from the website in zip format (.7z). Then, they are converted to .gzip format before uploading to the HDFS cluster. Detail description of each data table is in table 1

An excerpt of raw data from the Posts table is shown below:

```
s2812940@wegdam:~$ hdfs dfs -text /user/s2812940/
    project/data/Posts.gz | head -3
<?xml version="1.0" encoding="utf-8"?>
<posts>
  <row Id="4"
PostTypeId="1"
AcceptedAnswerId="7"
CreationDate="2008-07-31T21:42:52.667"
Score="787"
ViewCount="69172"
Body="&lt;p&gt;I want to assign the decimal
    variable &amp;quot;trans&amp;quot; to the
    double variable &amp;quot;this.Opacity&amp;
    quot;.&lt;/p&gt;&#xA;&lt;pre class=&quot;lang-
    cs prettyprint-override&quot;&gt;&lt;code&gt;
    decimal trans = trackBar1.Value / 5000;&#xA;
    this.Opacity = trans;&#xA;&lt;/code&gt;&lt;/
    pre&gt;&#xA;&lt;p&gt;When I build the app it
    gives the following error:&lt;/p&gt;&#xA;&lt;
    blockquote&gt;&#xA;&lt;p&gt;Cannot implicitly
    convert type decimal to double&lt;/p&gt;&#xA;&
    lt;/blockquote&gt;&#xA;"
OwnerUserId="8"
LastEditorUserId="16124033"
LastEditorDisplayName="Rich B"
LastEditDate="2022-09-08T05:07:26.033"
LastActivityDate="2022-09-08T05:07:26.033"
Title="How to convert Decimal to Double in C#?"
Tags="&lt;c#&gt;&lt;floating-point&gt;&lt;type-
    conversion&gt;&lt;double&gt;&lt;decimal&gt;"
AnswerCount="14"
CommentCount="7"
FavoriteCount="0"
CommunityOwnedDate="2012-10-31T16:42:47.213"
ContentLicense="CC BY-SA 4.0" />
```

Listing 1. Excerpt of raw data from Posts table. The schema from excerpt data is used in importing XML files later.

All data tables are in the self-closing XML format, meaning that they are non-splittable. Thus, when importing these data, we need to use an additional library spark-xml, version 2.11-0.9.0.

The combination of non-splittable format and large data size makes importing process become time-consuming. We can speed up the importing process significantly by passing a pre-defined schema when importing XML files (Please see the script for more detail).

Spark-xml allows reading XML files in the local or distributed filesystem as Spark DataFrames. Thus, after importing successfully, we save all data frames to parquet format (in 10 partitions) to reuse for analysis. Reading big data in parquet files is much faster than other formats.

3.2 Data analysis

This section presents the data analysis approaches to answer three research questions. We combine machine learning techniques and visualization analysis to answer questions.

- RQ 1: Clustering & Visualization
- RQ 2.1, 2.2: Visualization
- RQ 2.3: Sentiment Analysis, Feature Engineering (PCA) & Visualization

3.2.1 Tags clustering. According to the schema of Stack Exchange Data Explorer, tags can be summarised based on three tables: PostTags, Tags, and TagSynonyms. However, only table Tags are available in the archived data. Missing these important tables poses two challenges for answering the research question. First, we could not link Tags directly to the Posts table because of the missing PostTags table. Second, tags with similar meanings can not be grouped because of the missing TagSynonyms table.

For the first challenge, we decide to reproduce aggregation for tags from the Posts table and use the Tags table as a reference (for the frequency of tags). For the second challenge, we perform clustering to group tags with similar topics. This section will discuss two approaches used for clustering and the result.

Raw tags can be extracted from column "_Tags" in the Posts table. Only questions (PostTypeId = 1) contain tags. After cleaning the tags, we extract a total of 69,111,608 tags from these questions, with 64,155 unique tags. For each unique tag, we count the times it appears in the questions. We also calculate the cumulative frequency of these tags in order to reduce the data size. Indeed,

We consider two approaches for clustering the tags, namely, K-means clustering and Affinity Propagation. For the first approach, K-means clustering, we need to choose the number of clusters (k) in advance. In this experience, we try two settings: k = 10 and k = 20, and apply the K-means algorithms on the corpus of 69,111,608 raw tags. Raw tags are converted to features using the TF-IDF vectorization algorithm and then applying K-means for clustering.

For the second approach, Affinity Propagation, we do not need to choose the number of clusters as it is automatically generated from the algorithm. For this approach, we use the input of 4,748 unique tags as they account for 90% of the tags in SO questions.

The results of both approaches are discussed in the next section. Clustering results are used to assess whether we should analyze technology trends based on separate tags or based on tag clusters.

3.2.2 Text pre-processing.

The original data from StackExchange are in HTML format and contain different unused information. Therefore, the text must be pre-processed for Sentiment Analysis and Features Engineering. First, the code paragraph is extracted from the original text, and then all the `HTML` blocks and other non-readable characters are removed.

3.2.3 Sentiment analysis for posts content.

We performed Sentiment analysis for each post (excluding the code lines) using the Valence Aware Dictionary and sEntiment Reasoner (VADER) algorithm [15] built into the Natural Language Toolkit (NLTK) package in Python [5]. VADER is a simple rule-based algorithm for text sentiment analysis built on a lexical approach - mapping words to different sentimental categories (positive, neutral, and negative) with scores for each. For a sentence (and paragraph), the sentiment value is calculated by summing up values from each word and normalizing them to -1 to 1. The three values for each category will be normalized again in the range of $(-1, 1)$, which gives the final compound sentiment score. An additional feature of VADER, apart from a lexical-emotion dictionary, is heuristics. VADER can capture more than just words' emotion, including punctuation, capitalization, degree modifiers, polarity shift with "but", and polarity negation.

3.2.4 Feature Engineering.

For PCA, we mined the text and engineered different features based on word count, sentence count, character count, and syllable count, in addition to sentiment score (mentioned in section 3.2.1 above).

Feature 1: Coleman-Liau index (CLI)
The CLI scores the readability and understandability of a text, which will output a score approximated to the U.S. grade level [8] - for example, a grade of 12 denotes that the text is appropriate for a final year U.S. high school student. The calculation is as follows:

$$CLI = 0.0588L - 0.296S - 15.8 \qquad (1)$$

With L denotes the average number of letters per 100 words, and S denotes the average number of sentences per 100 words.

Feature 2: Flesch Reading ease (FRE) The Flesch Reading ease test was developed by Rudolf Flesch in the early 1940s [10]. The scores are also mapped to U.S. grade level, albeit with a different scale shown in table 2 below:

Score	Grade Level
90 to 100	5th Grade
80 to 90	6th Grade
70 to 80	7th Grade
60 to 70	8th and 9th Grade
50 to 60	10th to 12th Grade (high school)
30 to 50	college
0 to 30	college graduate

Table 2. Flesch Reading ease score mapping.

The formula for Flesch Reading ease is:

$$FRE = 206.835 - (1.015 * \frac{W}{Sent}) - (84.6 * \frac{Syll}{W}) \qquad (2)$$

With $Sent$ denotes the sentence count, W is the number of words in the paragraph, and $Syll$ is the syllable count. While FRE has an upper bound (approximately 121.22, where every sentence contains only one one-syllable word), it does not have a lower bound. Therefore, the score could reach as low as possible by using multiple-syllable words. This means FRE places more emphasis on syllables compared to other reading ease metrics such as CLI.

Feature 3: Flesch-Kincaid grade (FKG)

The Flesch-Kincaid grade is an improvement upon the original Flesch Reading ease, developed by J. Peter Kincaid for the U.S. Navy [16], which can directly output to the U.S. grade level instead of a conversion table. The calculation is as follows (with the same notations as equation 2):

$$FRE = 0.39 * \frac{W}{Sent} + 11.8 * \frac{Syll}{W} - 15.59 \quad (3)$$

Feature 4: Percentage of character inside HTML code block (code percentage)
This metric is straightforward: We calculated the number of characters inside the HTML `<code>*<\code>` block, and divide it by the total amount of characters in the paragraph.
Feature 5: Cosine similarity between title and tags of a post
Cosine similarity is defined as the similarity between two sequences of numbers, in which each sequence is modelled as a vector in an inner product space, and cosine similarity is the cosine angle between the two vectors - shown in equation 4:

$$S_c(V_1, V_2) = \frac{V_1 \cdot V_2}{||V_1||||V_2||} \quad (4)$$

With S_c denoting Cosine similarity, V_1 and V_2 are the two vectors. As Cosine similarity deals with vectors, we use the term frequency-inverse document frequency (TF-IDF) to convert a paragraph/sentence/a string of texts (in this case, tags and titles) into a vector that denotes how important terms are to the texts. TF-IDF can be calculated by multiplying the term frequency and the inverted document frequency.
Feature 6: Cosine similarity between body and title of a post
This feature is similar to feature 5, but here we calculated the Cosine similarity between the body and title.
Helper features: Sentence count, word count, and character count
These features are defined as 'helper' since it is not taken into account in PCA. However, they are used in the calculations of other features. Word count and character count are done using native Python's `split` and `len` methods, respectively. To determine the sentence count, we use `NLTK`'s `sent_tokenize` method, which uses Punkt tokenizer algorithm [17] to determine end-of-sentence.

3.2.5 Principal Component Analysis on Features.

Both linear dimensionality reduction techniques are utilized in determining the factors contributing to a question being answered, as posed by RQ 2.3. To answer the research question, we first label _all questions in the dataset in 3 distinct categories, namely: *Accepted Answer; Answered; Unanswered.* We mainly analyze the post content, i.e., the Body and Title of the question and numerical values as the Score(sum of upvotes & downvotes). As text comparison on large data is expensive, we perform feature engineering on the dataset described in the section above. We hypothesize that these features, or a combination thereof, could be a significant factor that leads to a question being answered. All questions are analyzed in a script that transforms the data into their feature values and writes this to a $\approx 1GB$ file. The years investigated were from 2015 to 2022, by choice, as we only managed to get PCA working in local mode and therefore had size limitations.

Firstly, a correlation matrix is plotted on the features to see if there is a basis for dimensionality reduction. Then, a Principal Component Analysis is performed on the extracted features, where we only input the features, and thus not the labels. Since all features coincide in different ranges of values, we use PySparks `StandardScaler` to normalize the features, which are then put into PySpark's PCA and run on the cluster, with the number of components set to $k = 7$. We have chosen this number of components since an initial investigation shows a moderately weak almost correlation between all features. Finally, we visualize the explained variation of the principal components and a selection of transformed data points on the top 2 principal components.

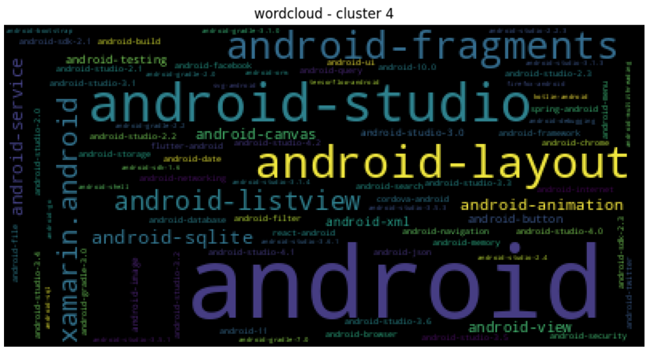


Figure 1. Wordcloud generated from K-means algorithm with 20 clusters - Cluster 4 - Theme: Android

4 Results

4.1 Tags clustering

4.1.1 Tags Clustering using K-means algorithm.

The results obtained from K-means clustering with 20 clusters are better than those with 10 clusters. Most clusters represent programming languages such as javascript, python, java, android, etc. Some examples of clusters using wordcloud visualization.

However, there is one cluster containing mixed terms of different languages, including tags with high frequency. We extract the top 20 tags in frequency to compare with clustering results.

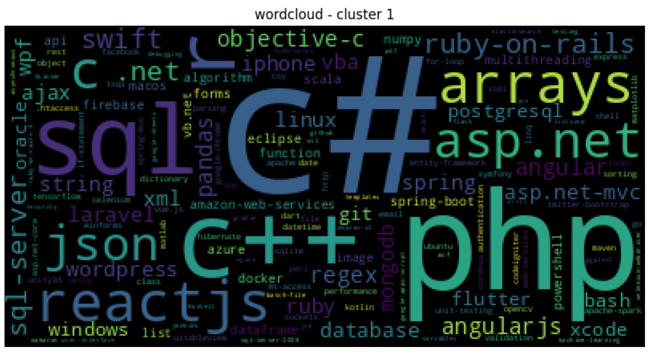


Figure 2. Wordcloud generated from K-means algorithm with 20 clusters- Cluster 1 - Mixed theme of different languages

4.1.2 Tags Clustering using Affinity Propagation algorithm.

4,748 unique tags (accounting for 90% of tags) were used as input of affinity propagation algorithm with different settings to generate clusters. Although using "euclidean" affinity generates much fewer clusters than "pre-computed" affinity (119 clusters vs 520 clusters), the output does not reveal meaningful semantic insights. Indeed, tags with similar spellings are more likely to be grouped together. Below are some exemplars (clusters) from this algorithm. Below are two excerpts of clusters with high and low semantic relationships:

Conclusion: To sum up, using clustering at the tag level is not sufficient because of the limitation in the semantic relationship between these technical terms (such as programming languages). Although the K-means algorithm results in better clusters than Affinity Propagation, it still cannot handle semantic limitations in one cluster containing mixed tags of different programming languages. Thus, we decide to use tags themselves rather than tag clusters for analyzing technology trends.

4.2 Technology trends reflection in Tags

To understand the technology trends reflected in the tags, we visualize the frequency of top popular tags over the last eight years (2015 - 2022). To capture the possibility of emerging technology (in less popular tags), we do visualization at three levels of popularity: Top 20, top 20-40, and top 40-60. For high-resolution figures, please refer to the appendix.

As can be seen from figure 3 Python, JavaScript, and Java are the three most commonly discussed languages in the last eight years. Based on the frequency of these tags in SO questions, we can see that Python is becoming more popular. The tag "Python" has taken over the first place of popularity from "JavaScript" since 2018. Both tags "Java" and "JavaScript" see a decreasing trend in their frequency, but Java decreases faster. The year 2019 also marks a sharp decrease in the tag

Table 3. Examples of Affinity propagation clusters

High semantic relationship	**Low semantic relationship**
google-maps-api-3 google-admin-sdk, google-analytics, google-api-client, google-app-engine, google-apps-script, google-chrome-app, google-cloud-run, google-cloud-sql, google-data-studio, google-drive-api, google-maps-api-2, google-maps-api-3, google-places-api, google-play-games, google-sheets-api, google-tag-manager, google-workspace	**jupyter-notebook**: active-directory, apache-zookeeper, asp.net-core-2.0, asp.net-core-2.1, asp.net-core-3.1, asp.net-core-mvc, asp.net-web-api2, azure-databricks, azure-powershell, buffer-overflow, cluster-analysis, complexity-theory, computer-science, copy-constructor, create-react-app, custom-post-type, entity-framework, facebook-ios-sdk, flutter-provider, full-text-search, game-development, hibernate-search, ipython-notebook, jquery-selectors, jquery-ui-dialog, jupyter-notebook, laravel-passport, numerical-methods, observer-pattern, operator-keyword, phaser-framework, python-itertools, python-tesseract, react-router-dom, react-typescript, restructuredtext, simpledateformat, spring-boot-test, spring-websocket, ternary-operator, youtube-data-api
android-alertdialog: android-4.4-kitkat, android-alertdialog, android-custom-view, android-databinding, android-mediaplayer, android-permissions, android-preferences, android-progressbar, android-scrollview, android-studio-3.0, android-tablelayout, android-workmanager, angular2-directives, undefined-behavior.	**require**: adsense, archive, aurelia, clojure, designer, eclipse, execute, gesture, getline, gremlin, ireport, metrics, ncurses, newline, openfire, ormlite, perforce, predict, preview, primeng, profile, profiler, promise, qr-code, reactive, reactjs, readfile, readline, redirect, redmine, reduce, refresh, registry, release, rename, repaint, repeater, request, require, reshape, resize, response, restart, restkit, restlet, restore, retrofit, return, reverse, rewrite, runtime, security, serilog, service, version, vertica

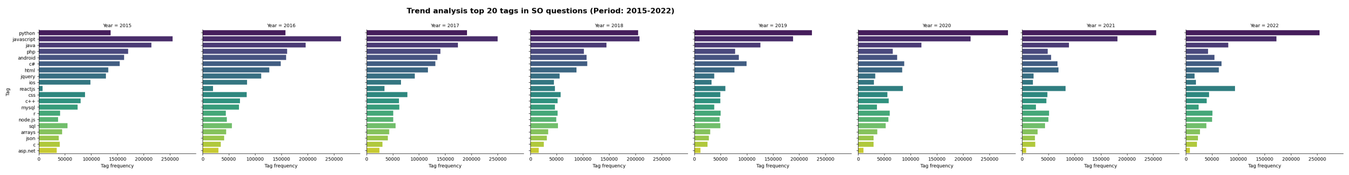


Figure 3. Technology trend reflection in tags - Frequency in SO questions - Top 20 popular tags (Period: 2015-2022) - View high-resolution figure here

frequency of other programming languages such as PHP, Android, C#, and HTML by almost 50%.

The arising popularity of Python is confirmed by two tags, "python-3.x" and "pandas" (a library of Python), in the top 20-40 tags (Figure 4). Both terms appear more frequently in SO questions during the last eight years. Another interesting technology trend captured in this figure is the replacement of Google's popular framework AngularJS with Angular. Since the release of Angular in 2016[2], this term has gradually substituted for "angular" in SO questions.

In the last popularity level (top 40-60th tags, figure 15), the term "flutter" (an open-source UI software development kit created by Google, released in 2017) has gained popularity more than five times in 5 years since its release. The tag "dataframe" also has a notable increase in frequency in SO questions.

We also filter tags relating to Spark and visualize its change in the same period, which is shown in figure 16. "apache-spark" is the most frequently used among these tags, followed by "pyspark" and "apache-spark-sql". "pyspark" catches up quickly with "apache-spark", both tags frequently appear in SO questions as at 2022.

4.3 Influence of Tags on the possibility of getting answered

Using visualization, we aim to answer whether tags can influence the possibility of getting answers for SO questions regarding quantity (number of tags) and quality (the content of tags).

4.3.1 Number of tags per question.

SO questions are aggregated based on the number of tags per question and the possibility of getting answered. As can be seen in Figure 5, most of the questions in Stack Overflow have two or three tags and a maximum of 6 tags (though it is rarely seen). However, the number of tags does not influence how likely it is to get answered. The percentage of getting answered is approximately 50/50 across six categories.

[2]https://www.pangea.ai/angular-resources/angular-vs-angularjs-what-are-the-differences/

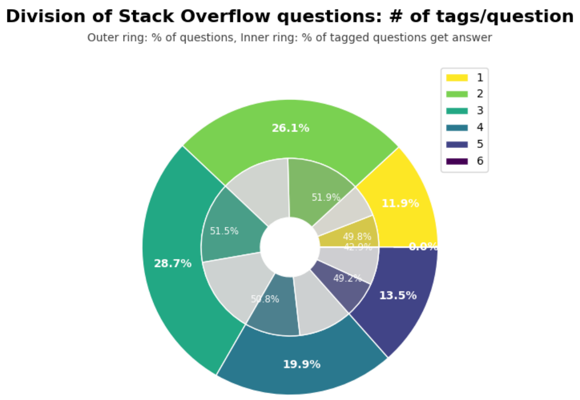


Figure 5. Correlation between number of tags per question and the possibility of getting answered

4.3.2 Adding popular tags to questions.

As can be seen from Figure 18, adding popular tags (top 20) does not guarantee a high chance of getting an answer. Most of these terms share the same medium possibility (50-60%). However, adding these terms to the questions may reduce the waiting time to get answered (Figure 6). We measure waiting time by taking the difference between the creation time of the question and its corresponding answer. According to the figure, the waiting time for the top three popular tags ("python", "java", "javascript") has lower waiting time than the bottom three tags (asp.net, .net, ruby-on-rail) in the list. Questions about "iOS", "Android", and "Node.js" seem to be equally challenging with long waiting times, regardless of how frequently the tags are added to these questions.

4.4 Influence of user reputation on the quality of posts

To answer research question 2.2, we explored the Pearson correlation between user reputation and the quality of Stack Overflow posts. In accomplishing it, we examined three key aspects of this relationship: the correlation between user reputation and badges, the correlation between user reputation and the quality of answers, and the correlation between user reputation and the quality of questions. By understanding

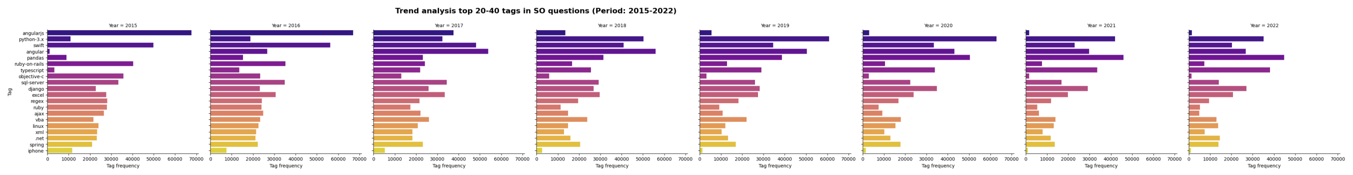


Figure 4. Technology trend reflection in tags - Frequency in SO questions - Top 20-40 popular tags (Period: 2015-2022). "python-3.x" and "pandas" become more popular. View high-resolution image here.

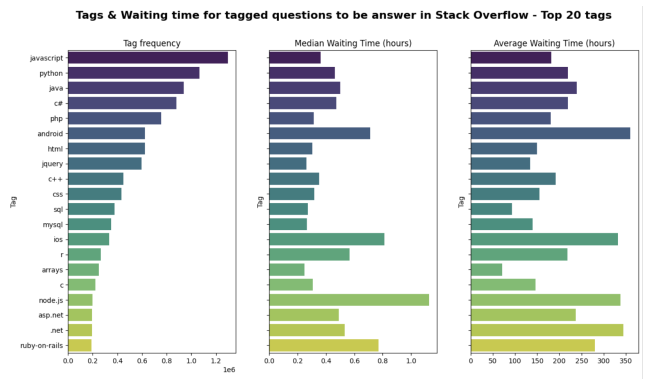


Figure 6. Correlation between the content of tags and the waiting time to be answered - Top 20 tags

these correlations, we gained insights into whether user reputation is a valid indicator of post quality on Stack Overflow.

4.4.1 Correlation between user reputation and badges.

Badges are a visual representation of a user's achievements and contributions to the platform, and they could be used as a measure of a user's expertise on the site. Badges were earned by performing specific actions, such as asking and answering questions, receiving upvotes, and participating in specific events. Examining the correlation between user reputation and badges provided insights into whether user reputation was an accurate reflection of a user's expertise on Stack Overflow.

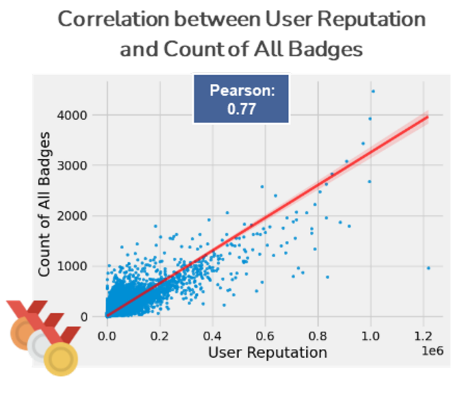


Figure 7. Correlation between User Reputation and Count of All Badges

Figure 7 revealed a strong positive correlation between user reputation and the count of badges on Stack Overflow, with a Pearson correlation coefficient of 0.77. Furthermore, the correlation was also analyzed for each count of bronze, silver, and gold badges, and the results showed strong positive correlations for all three types of badges. Moreover, the results show a particularly strong correlation between user reputation and silver badges, with a coefficient of 0.83. The detailed correlation visualization for each count of badges can be found in Appendix C. These findings indicate that user reputation is a reliable indicator of a user's helpfulness on the platform, although as mentioned in [27], it might not fully reflect the user's expertise. However, as both are strongly correlated, we decided to move to the next subsections using user reputation as a metric to examine if there are any correlations between user reputation and the quality of posts on Stack Overflow.

4.4.2 Correlation between user reputation and quality of answers.

In order to assess the relationship between user reputation and the quality of answers on Stack Overflow, two Pearson correlations were calculated and visualized. The first correlation measured the relationship between user reputation and the count of accepted answers. It was assumed that users with higher reputations are more likely to provide answers that are deemed the best by the asker, thus having higher counts of accepted answers. The second correlation measured the relationship between user reputation and the sum of answer scores, where the score was defined as the difference between the number of upvotes and downvotes. This second correlation aimed to assess whether higher reputation users provide answers that are more highly rated by the community. Figure 8 highlights a very strong positive correlation between user reputation and the quality of answers, with a Pearson correlation coefficient of 0.80 between user reputation and count of accepted answers, and a Pearson correlation coefficient of 0.96 between user reputation and the sum of answer scores. These findings support that user reputation is a good indicator of the quality of answers given on the Stack Overflow platform.

4.4.3 Correlation between user reputation and quality of questions.

This section aims to comprehend the relationship between

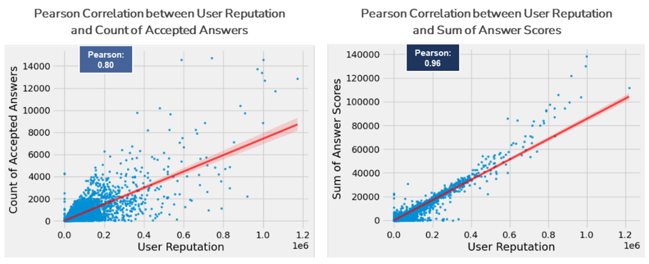


Figure 8. Correlation between User Reputation and Quality of Answers

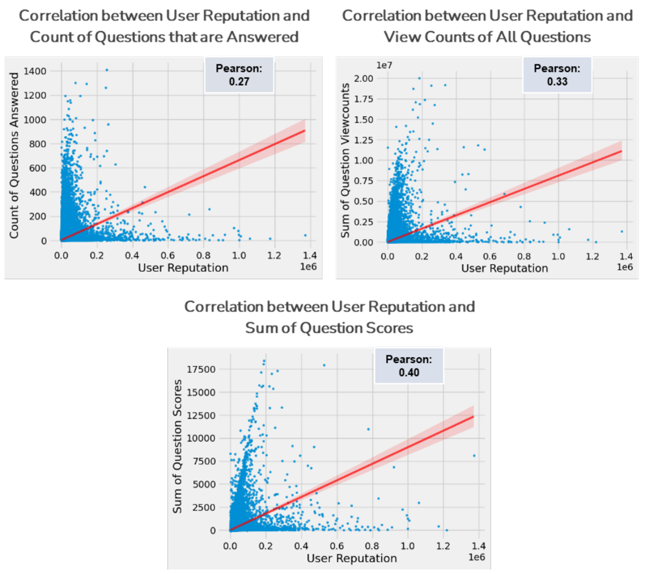


Figure 9. Correlation between User Reputation and Quality of Questions

user reputation and the quality of questions asked on Stack Overflow. We broke down the term quality of questions into three Pearson correlations. The first correlation was between the user reputation and the count of answered questions. Only the questions which other users answered were considered. The second correlation was between the user reputation and the view counts of all questions asked by the user. Lastly, we calculated the correlation between user reputation and the sum of scores for the questions, where the score was defined as the number of upvotes minus the number of downvotes.

The relationship between user reputation and the quality of questions on Stack Overflow is visualized in Figure 9 above. The results showed weak positive correlations between user reputation and the count of answered questions (0.27) and view counts of all questions (0.33). Moreover, the correlation between user reputation and the sum of question scores was relatively moderate (0.40). Interestingly, it can be deduced that users with higher reputations are more likely to provide high-quality answers than ask questions on Stack Overflow. Contrarily, users with lower reputations, especially around 200,000, have the highest count of answered questions, view counts, and question scores.

4.5 How fast does a question get answered?

In this section, we take a look at the time between a question being posted and the time when an answer is marked as "accepted" by the account that asked the question. The results are shown in figures 10 below, and it is surprising: Out of a relatively moderate sample that we tested (1.2 million questions with an answer marked as accepted), the majority of it (70%) has an accepted answer within one hour of posting. Almost 100% of the questions have an accepted answer within one day of posting. The longest "waiting time" for an answer is 4,620 days, which translates roughly to 12 years and seven months. This highlights the ability of Stack Overflow as a database of questions and answers, where a question asked by someone today might not be answered yet (or an answer might be outdated). Still, in the future, when more people encounter problems, different answers will arise.

4.6 Sentiment Analysis of post contents

Figure 11 shows the sentiment compound score plotted against the number of answered questions. Overall, the majority of questions (73%) have either a neutral or positive sentiment score, and 43% have a very positive sentiment score. This shows an interesting insight that most questions are toned positive or very positive. The hypothesis here is that users would associate positively-toned questions with faster response time. However, when calculating the correlation matrix, it shows that there is no correlation between waiting time (of questions that already have an accepted answer) and the sentiment score of the questions.

4.7 PCA

Figure 12 shows the correlation amongst all the features chosen. There is a clear negative correlation between the Fresch-Kincaid Grade and the Fresch Reading Ease. Furthermore, the Fresch Reading Ease is also negatively correlated with the Coleman-Liau Index. In contrast, the Fresch Grade and Coleman-Liau index seem highly correlated. All other features in the matrix have less than ±0.1 difference in correlation.

The results for the PCA analysis can be found in figure 13. The plot shows that the two principal components need to be selected to explain half of the data. However, we still need to select multiple principal components to capture the information in all the data. This is possibly due to the low correlation between the presented features. The above is furthermore supported when looking at figure 14, as this shows the data transformed on the eigenvectors of the top two principal components. As can be seen from this graph, is

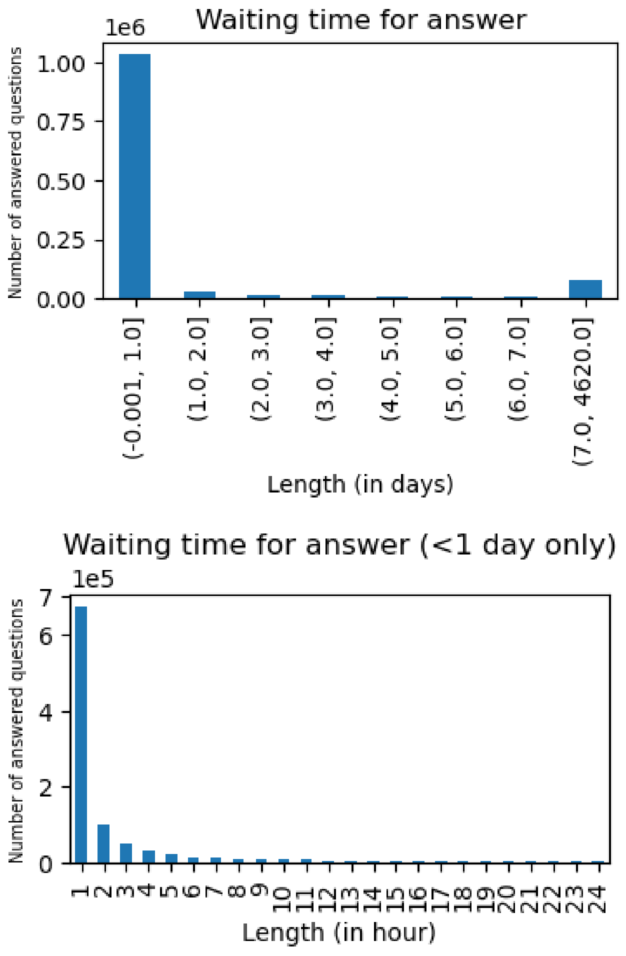


Figure 10. Time between question posted and answer accepted

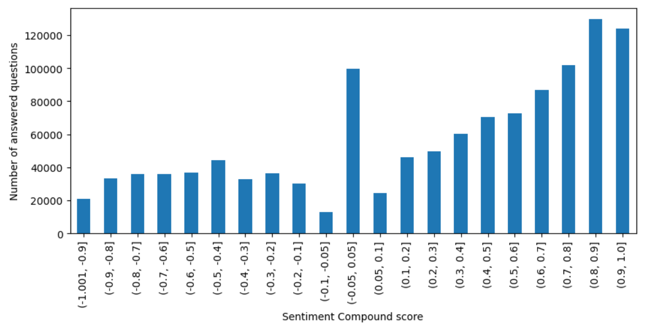


Figure 11. Sentiment compound score for posts with accepted answer

that the data points still lie closely together and lack discriminative power. This aspect occurs throughout all principal components and suggests the relationships between these features may be non-linear.

5 Discussion and critical reflection

5.1 On RQ1.1 & RQ2.1

Tags in SO questions mainly capture programming languages used by the users (developers) when they post their questions rather than the specific issues they are facing.

Tags themselves may represent the changes in technology trends through the rise and fall in frequency of a tag in SO questions. The replacement of Google's AngularJS with Angular is one example. Our findings about dominant programming languages such as Python, Java, and JavaScript are consistent with previous studies.[26]

We performed clustering at tag level (word) in order to capture technology topics being discussed, but the results were not as expected. The first possible cause is that we could not customize distance measures for K-means clustering as [4]. The second possible cause is the corpus should combine post content (Body) instead of only tags because tags only appear in questions and not in answers. Using language models on this corpus[25], community detection[11] [3] or graph-based approach could be considered in future research.

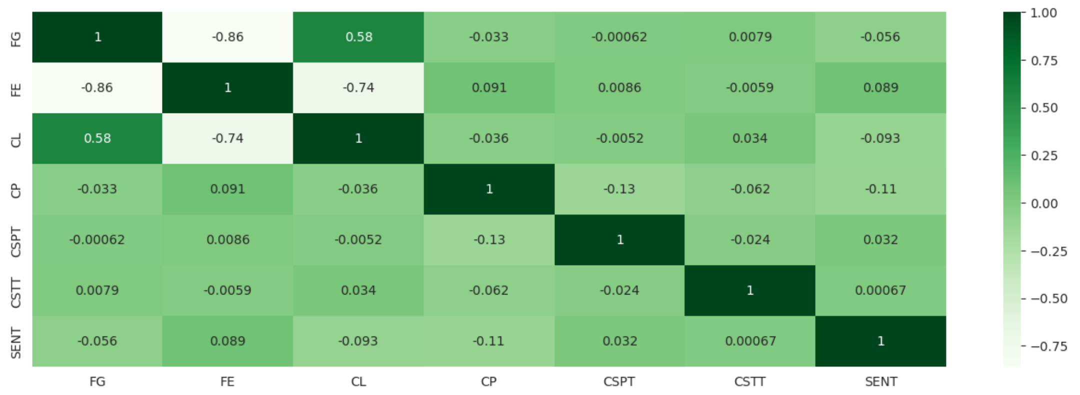
Finally, the visual analysis highlights the importance of tag content rather than the number of tags in questions on the possibility of getting an answer. Specifically, putting more tags in the question does not necessarily result in a higher chance of getting answers to the question. However, putting popular tags in the questions may reduce the waiting time for getting a response.

5.2 On RQ2.2

The results of the correlation analysis in Section 4.4.2 and 4.4.3 show that user reputation is more strongly correlated with the quality of answers on Stack Overflow rather than with the quality of questions. Several factors can explain this phenomenon: The user reputation system in Stack Overflow is designed to acknowledge and reward users for their contributions to the platform. Stack Overflow places a significant emphasis on having precise and helpful answers. Hence, users who consistently provide high-quality answers are likely to be recognized by the community, and their answers are upvoted and accepted. This leads to higher reputation scores for those users.

In contrast, the quality of questions asked by a user is not given as much importance as the quality of answers. This is because the main focus of the platform is to provide answers to questions rather than to ask questions. As a result, users who consistently ask high-quality questions are not likely to receive as much recognition from the community. Their questions are not likely to receive as many upvotes. Hence, this results in a weaker correlation between user reputation

[3]An inspiring visualization to cluster SO tags: https://twosixlabs.github.io/visualizing-tags/



	FG	FE	CL	CP	CSPT	CSTT	SENT
FG	1	-0.86	0.58	-0.033	-0.00062	0.0079	-0.056
FE	-0.86	1	-0.74	0.091	0.0086	-0.0059	0.089
CL	0.58	-0.74	1	-0.036	-0.0052	0.034	-0.093
CP	-0.033	0.091	-0.036	1	-0.13	-0.062	-0.11
CSPT	-0.00062	0.0086	-0.0052	-0.13	1	-0.024	0.032
CSTT	0.0079	-0.0059	0.034	-0.062	-0.024	1	0.00067
SENT	-0.056	0.089	-0.093	-0.11	0.032	0.00067	1

Figure 12. Correlation matrix of features on questions.

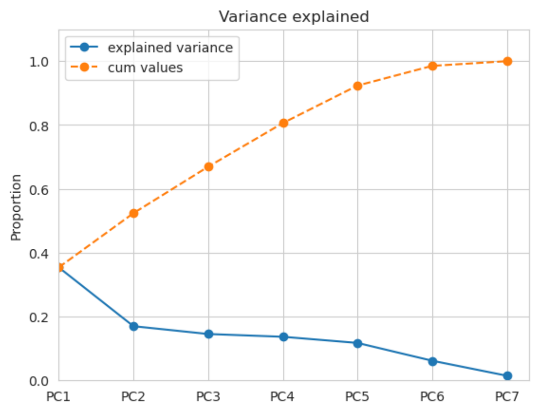


Figure 13. Principal Component Analysis on presented features.

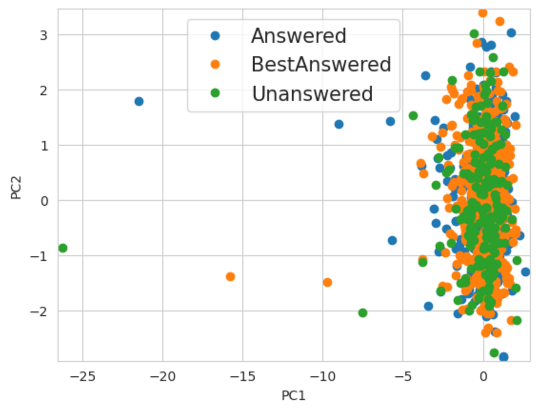


Figure 14. Plotting PC1 vs. PC2 on sample of the (scaled) features.

and the quality of questions asked by the user.

5.3 On RQ2.3

Feature Engineering and Sentiment Analysis were a challenge since external libraries have to be used (`NLTK`, `scikit-learn`). Therefore it has to be done row-by-row using `Pyspark UDF`. This makes the program extremely slow, which means features are only engineered on 1.2 million rows of data. Different approaches have been tested, including the `Pandas-UDF`; however, due to the time constraint, the Sentiment Analysis and, subsequently, PCA is only available for 1.2 million rows and 13 million rows, respectively (there are 57M questions in the whole dataset).
For different features, there are also multiple limitations, namely:

- Sentence count: There are no definitive measures to calculate the number of sentences, as there are multiple variants to how an end-of-sentence is - especially on an informal (in terms of language) exchange like Stack Overflow with multiple grammatical errors. This makes counting the number of sentences challenging.
- Code percentage: Since "code" and "text" are formed in a particularly different way, it is difficult to use one metric (i.e., number of characters, number of lines) to compare both code and text. Hodgins [14] uses lines of code per total line of text. However, there is a major flaw in this method: The information density of text is much higher compared to code since code can be formatted, spaced, and split into different lines for readability. Therefore, we opted for comparing the characters of codes over the total characters.
- Readability metrics (CLI, FRE, and FKG): We acknowledge that there are multiple limitations: these metrics were developed for school books, and the formulas are only able to measure countable objects - human languages are much more than just numbers [24]. Another discussed limitation is that the metrics can be skewed for non-native English speakers. They might engineer a sentence in their mother tongue (which

can have longer sentences and different structures) and then directly translate it into English. This might lead to a score that denotes a hard-to-read paragraph for a native speaker (or speaker from other language families) but not for other non-native speakers that share the same language family.

- Sentiment score: There are also limitations with the VADER sentiment algorithm: Since it is not trained on Technology and Software texts, its performance might not be as robust when applied to Stack Overflow data. It is clearly seen in figure 11: there are multiple posts classified as "negative" or "extremely negative" (some even with scores -0.9 to -1). However, after excerpting those examples, the issues mainly lie in how the text seems straightforward and short, which makes the VADER algorithm classifies it as negative.

Given our Principal Component Analysis, we see that the selected features chosen in this paper are not easily distinguishable. The values are almost all uncorrelated, with the exception of the strange negative correlation between the Flesch Reading Ease(FE) and Flesche-Kincaid Grade(FG). This is strange because the FG builds further upon the FE, with the scale being inverted only. A reason for this could be the difference in magnitude of the scales of these features, where the FG lies between $[0, 100]$, and FE between $(-\infty, 121.22]$. However, the readability metrics FRE, FG and CL are all correlated nonetheless, which is logical.

The top 3 principal components only explain up to 70% of the variance in the dataset. Thus, it might be hard to remove features from the dataset without losing information. Note that PCA only analyses linear relationships between the data points. Other proven non-linear dimensionality reduction techniques could also have been used to discover non-linear relationships between the selected features, such as t-SNE.

6 Conclusion

In conclusion, this project focuses on two aspects of Stack Overflow, the change in technology topics being discussed and the quality of content.

First, analysis of the tags in Stack Overflow questions help reveals changes in technologies over time (last eight years), including technology replacement and the rise and fall of the popularity of programming languages. We find that Python has finally replaced JavaScript topics throughout the database. Moreover, we also see a decrease in other proven languages like Java, HTML, and PHP.

The second contribution of this project is to explain factors affecting the quality of posts in StackOverflow based on three aspects: tags (representation of questions), users' expertise, and the content of the post itself. Interestingly, tags perhaps do not strongly influence responsiveness. Adding more tags to a question, even when adding popular tags, does not necessarily increase the possibility of getting answers to your question. In contrast, we have confirmed the user's expertise to match the quality of the post, both answers and questions. Finally, we have investigated several features that determine the quality of the question by looking at the readability, sentiment, and code percentage of the question. Our PCA analysis has shown that there are no discriminative factors from our selected features that distinguish the question being answered or not. Furthermore, we found that answer time is typically within one hour, which in combination with our other results, may indicate the question being answered is a mere result of it being visible to the right people.

References

[1] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2012. Discovering Value from Community Activity on Focused Question Answering Sites: A Case Study of Stack Overflow. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Beijing, China) *(KDD '12)*. Association for Computing Machinery, New York, NY, USA, 850–858. https://doi.org/10.1145/2339530.2339665

[2] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2013. Steering User Behavior with Badges. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) *(WWW '13)*. Association for Computing Machinery, New York, NY, USA, 95–106. https://doi.org/10.1145/2488388.2488398

[3] Anton Barua, Stephen W. Thomas, and Ahmed E. Hassan. 2014. What Are Developers Talking about? An Analysis of Topics and Trends in Stack Overflow. *Empirical Softw. Engg.* 19, 3 (jun 2014), 619–654. https://doi.org/10.1007/s10664-012-9231-y

[4] Nartlada Bhakdisuparit and Iwao Fujino. 2018. Understanding and clustering hashtags according to their word distributions. In *2018 5th International Conference on Business and Industrial Research (ICBIR)*. 204–209. https://doi.org/10.1109/ICBIR.2018.8391193

[5] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

[6] Amiangshu Bosu, Christopher S. Corley, Dustin Heaton, Debarshi Chatterji, Jeffrey C. Carver, and Nicholas A. Kraft. 2013. Building reputation in StackOverflow: An empirical investigation. In *2013 10th Working Conference on Mining Software Repositories (MSR)*. 89–92. https://doi.org/10.1109/MSR.2013.6624013

[7] Fabio Calefato, Filippo Lanubile, Maria Concetta Marasciulo, and Nicole Novielli. 2015. Mining Successful Answers in Stack Overflow. In *Proceedings of the 12th Working Conference on Mining Software Repositories* (Florence, Italy) *(MSR '15)*. IEEE Press, 430–433.

[8] Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology* 60 (1975), 283–284.

[9] Daniel Hasan Dalip, Marcos André Gonçalves, Marco Cristo, and Pavel Calado. 2013. Exploiting User Feedback to Learn to Rank Answers in Q&amp;a Forums: A Case Study with Stack Overflow. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Dublin, Ireland) *(SIGIR '13)*. Association for Computing Machinery, New York, NY, USA, 543–552. https://doi.org/10.1145/2484028.2484072

[10] R. Flesch. 1979. *How to Write Plain English: A Book for Lawyers and Consumers.* Harper & Row. https://books.google.nl/books?id=-kpZAAAAMAAJ

[11] Casey Haber. 2021. Visualizing programming behaviors with Stack Overflow - Two six technologies: Advanced technology solutions for critical missions. https://twosixtech.com/visualizing-programming-behaviors-with-stack-overflow/

[12] Benjamin V. Hanrahan, Gregorio Convertino, and Les Nelson. 2012. Modeling Problem Difficulty and Expertise in Stackoverflow. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work Companion* (Seattle, Washington, USA) *(CSCW '12)*. Association for Computing Machinery, New York, NY, USA, 91–94. https://doi.org/10.1145/2141512.2141550

[13] Kerry Hart and Anita Sarma. 2014. Perceptions of Answer Quality in an Online Technical Question and Answer Forum. In *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering* (Hyderabad, India) *(CHASE 2014)*. Association for Computing Machinery, New York, NY, USA, 103–106. https://doi.org/10.1145/2593702.2593703

[14] Geoffrey Hodgins. 2016. Classifying the Quality of Questions and Answers From Stack Overflow.

[15] C. Hutto and Eric Gilbert. 2014. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media* 8, 1 (May 2014), 216–225. https://doi.org/10.1609/icwsm.v8i1.14550

[16] J. Peter Kincaid, Robert P. Fishburne, R L Rogers, and Brad S. Chissom. 1975. Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel.

[17] Tibor Kiss and Jan Strunk. 2006. Unsupervised Multilingual Sentence Boundary Detection. *Computational Linguistics* 32 (2006), 485–525.

[18] Bing Liu. 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies* 5, 1 (May 2012), 1–167. https://doi.org/10.2200/s00416ed1v01y201204hlt016

[19] Andrew Marder. 2015. Stack Overflow Badges and User Behavior: An Econometric Approach. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*. 450–453. https://doi.org/10.1109/MSR.2015.61

[20] Dana Movshovitz-Attias, Yair Movshovitz-Attias, Peter Steenkiste, and Christos Faloutsos. 2013. Analysis of the Reputation System and User Contributions on a Question Answering Website: StackOverflow. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (Niagara, Ontario, Canada) *(ASONAM '13)*. Association for Computing Machinery, New York, NY, USA, 886–893. https://doi.org/10.1145/2492517.2500242

[21] Nicole Novielli, Fabio Calefato, and Filippo Lanubile. 2018. A Gold Standard for Emotion Annotation in Stack Overflow. In *Proceedings of the 15th International Conference on Mining Software Repositories* (Gothenburg, Sweden) *(MSR '18)*. Association for Computing Machinery, New York, NY, USA, 14–17. https://doi.org/10.1145/3196398.3196453

[22] Nicole Novielli, Daniela Girardi, and Filippo Lanubile. 2018. A Benchmark Study on Sentiment Analysis for Software Engineering Research. *CoRR* abs/1803.06525 (2018). arXiv:1803.06525 http://arxiv.org/abs/1803.06525

[23] Maria Papoutsoglou, Georgia M. Kapitsaki, and Lefteris Angelis. 2020. Modeling the effect of the badges gamification mechanism on personality traits of Stack Overflow users. *Simulation Modelling Practice and Theory* 105 (2020), 102157. https://doi.org/10.1016/j.simpat.2020.102157

[24] Janice Redish. 2000. Readability formulas have even more limitations than Klare discusses. *ACM Journal of Computer Documentation* 24 (08 2000), 132–137. https://doi.org/10.1145/344599.344637

[25] Iman Saleh and Neamat El-Tazi. 2017. Automatic Organization of Semantically Related Tags Using Topic Modelling. In *New Trends in Databases and Information Systems*, Mārīte Kirikova, Kjetil Nørvåg, George A. Papadopoulos, Johann Gamper, Robert Wrembel, Jérôme Darmont, and Stefano Rizzi (Eds.). Springer International Publishing, Cham, 235–245.

[26] Sebastian Schuster, Wanying Zhu, and Yiying Cheng. 2013. Predicting Tags for StackOverflow Questions. https://cs229.stanford.edu/proj2013/SchusterZhuCheng-PredictingTagsforStackOverflowQuestions.pdf

[27] Shaowei Wang, Daniel M. German, Tse-Hsun Chen, Yuan Tian, and Ahmed E. Hassan. 2021. Is reputation on Stack Overflow always a good indicator for users' expertise? No!. In *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 614–618. https://doi.org/10.1109/ICSME52107.2021.00067

[28] Stav Yanovsky, Nicholas Hoernle, Omer Lev, and Kobi Gal. 2021. One size does not fit all: A study of badge behavior in stack overflow. *Journal of the Association for Information Science and Technology* 72, 3 (2021), 331–345. https://doi.org/10.1002/asi.24409 arXiv:https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.24409

[29] Jun Zhang, Mark S. Ackerman, and Lada Adamic. 2007. Expertise Networks in Online Communities: Structure and Algorithms. In *Proceedings of the 16th International Conference on World Wide Web* (Banff, Alberta, Canada) *(WWW '07)*. Association for Computing Machinery, New York, NY, USA, 221–230. https://doi.org/10.1145/1242572.1242603

Appendix A Script

All scripts and Jupyter notebooks for this project are available at this GitHub Repository

Appendix B Visualization analysis for tags

Below are additional figures relating to tags.

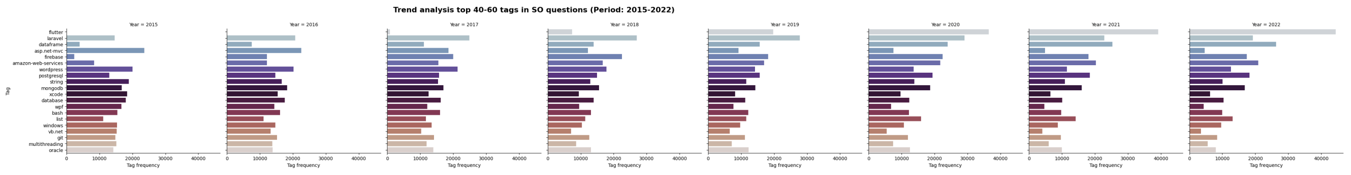


Figure 15. Technology trend reflection in tags - Frequency in SO questions - Top 40-60 popular tags (Period: 2015-2022). View high-resolution image here.

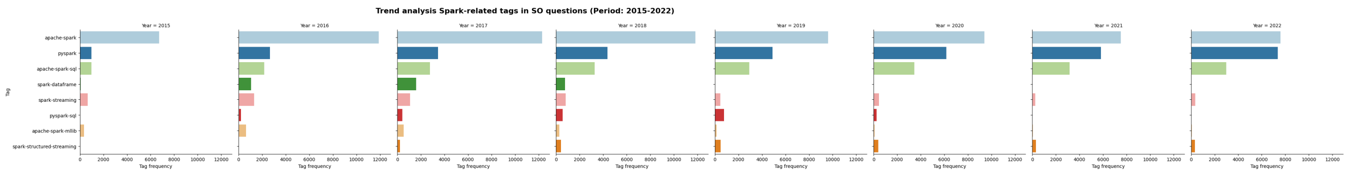


Figure 16. Trend analysis for Spark-related tags - (Period: 2015-2022). View high-resolution image here.

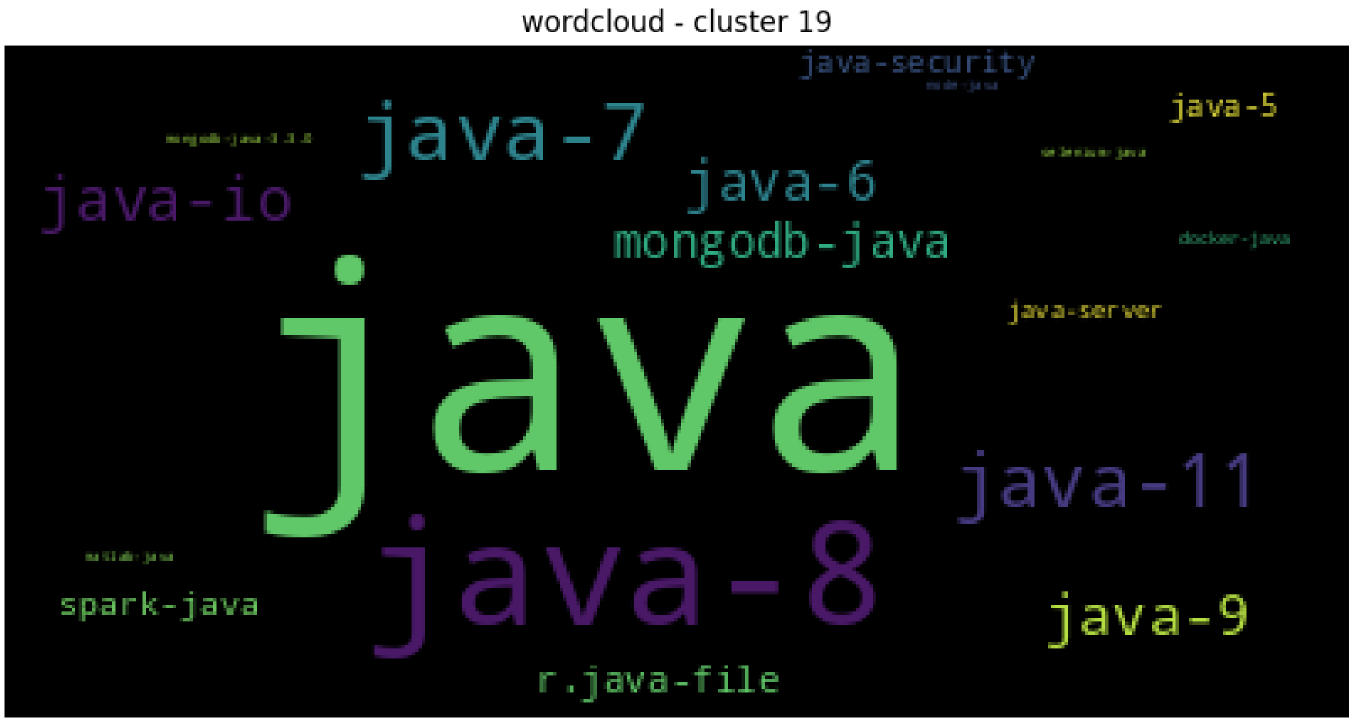


Figure 17. Wordcloud - Cluster 19 - java - Kmeans algorithm with 20 clusters

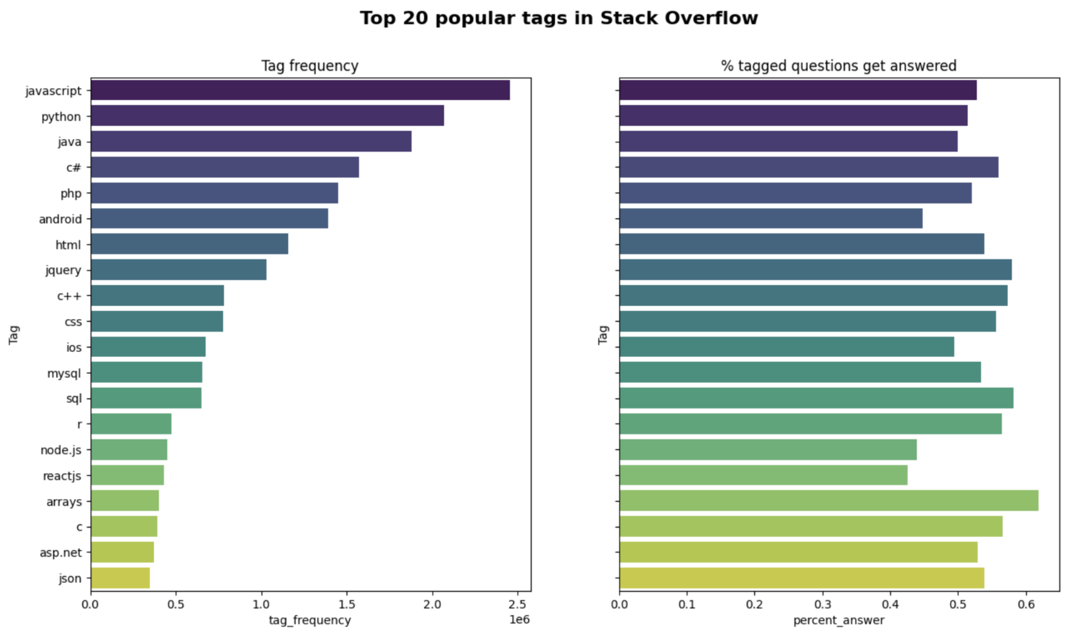


Figure 18. Correlation between the content of tags and the possibility of getting answers for SO questions

Appendix C Correlation between User Reputation and Badges

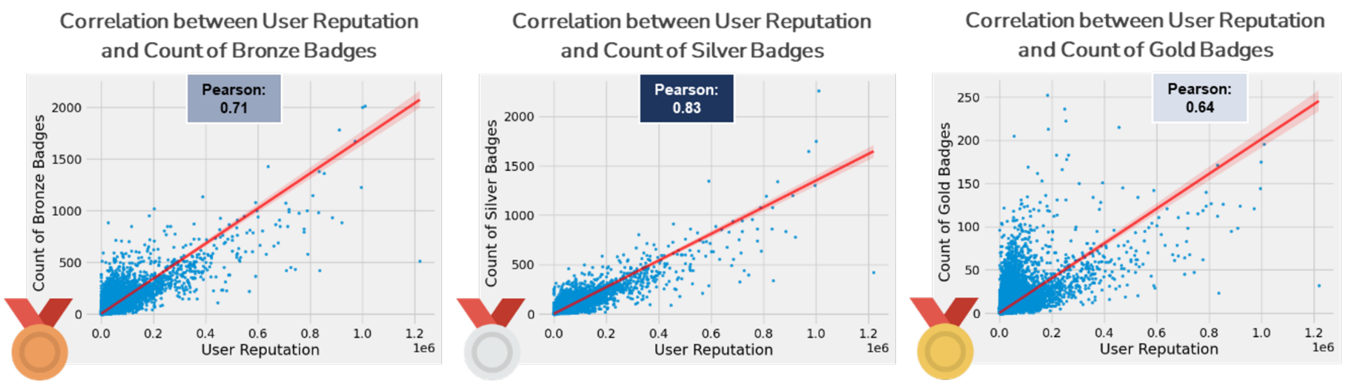
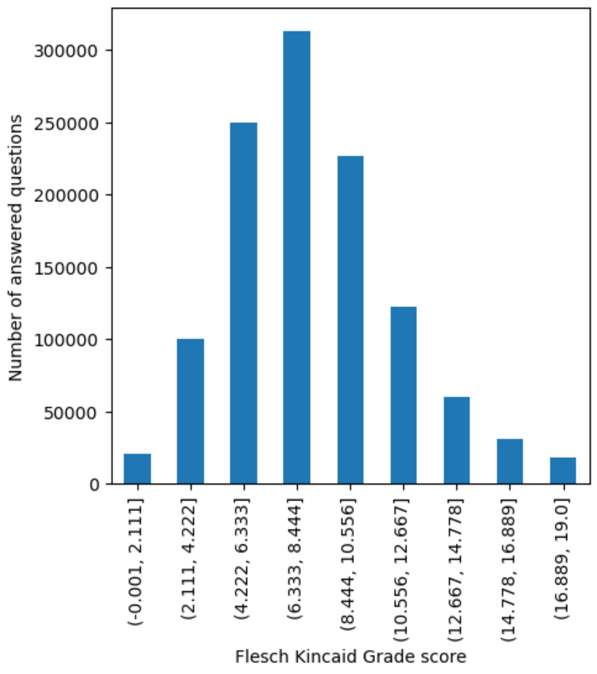


Figure 19. Correlation between User Reputation and Count of Bronze, Silver, Gold Badges

Appendix D Features Visualization



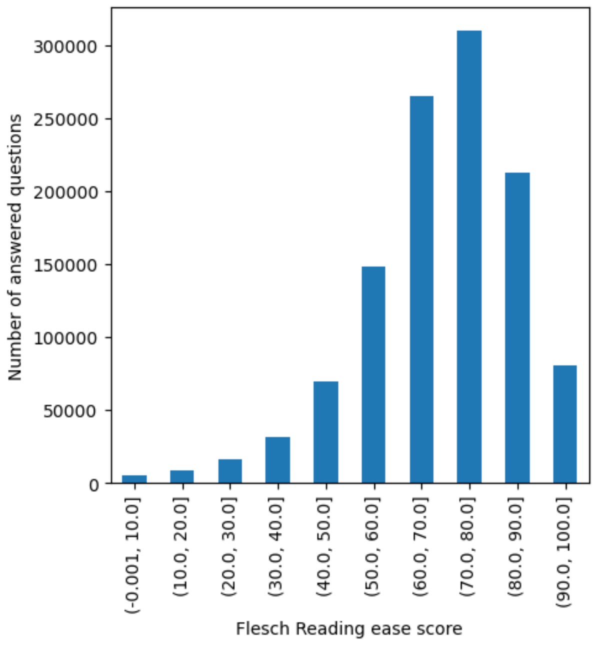
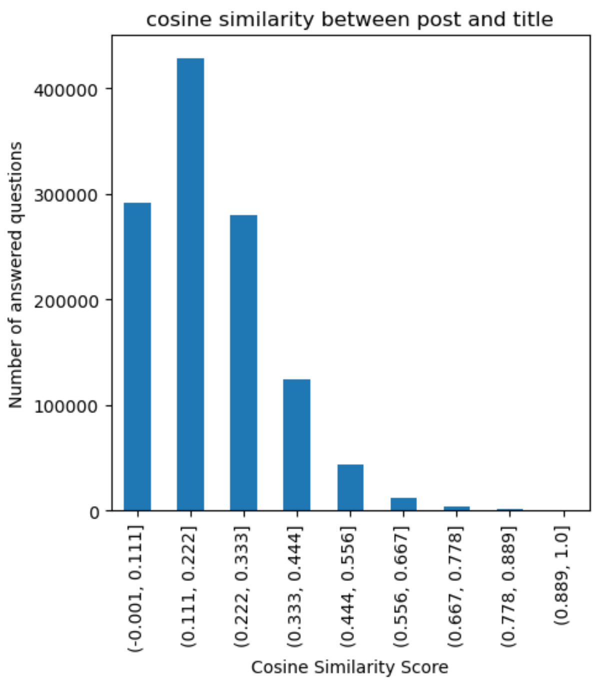


Figure 20. Flesch-Kincaid grade and Flesch Reading ease distribution



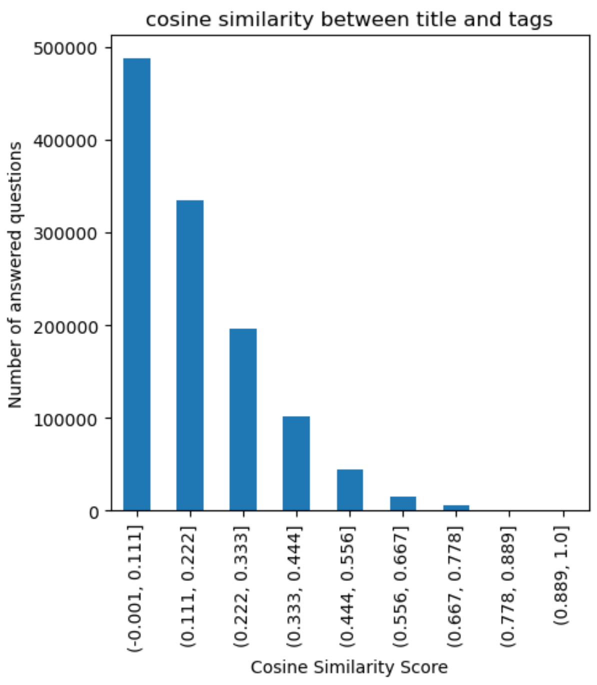


Figure 21. Cosine Similarity score between post-title and title-tag distribution

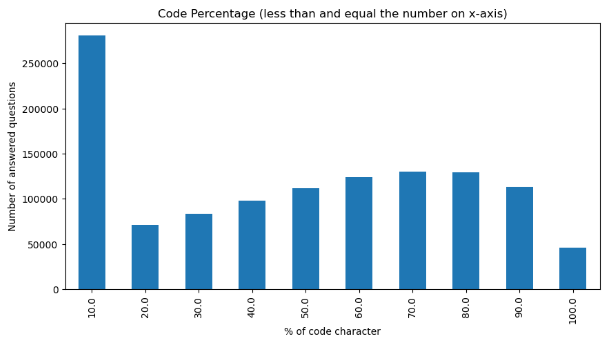


Figure 22. Percentage of codes in posts distribution