

TECHNISCHE UNIVERSITÄT
CHEMNITZ

Rise of JavaScript Frameworks - A Study on Stack Overflow

Master Thesis

Submitted in Fulfilment of the
Requirements for the Academic Degree
M.Sc.

Dept. of Computer Science
Chair of Software Engineering

Submitted by: Shyam Ashish Agrawal
Matrikel Nr.: 479026
Date: 18.01.2021

Supervisor: Prof. Dr. Janet Siegmund
M. Sc. Shadi Saleh

Abstract

Stack Overflow, a Q&A community, has become more prevalent among JavaScript developers to ask programming related questions to solve real-world problems. This study aims to determine the trending JavaScript frameworks in 2019 and how the Stack Overflow community promotes the learning of JavaScript frameworks. We analyzed the download statistics and number of questions asked on Stack Overflow for the top 3 JavaScript frameworks desired by the Stack Overflow community in their annual developer survey 2019. We conducted an experiment to classify the Stack Overflow data into different question categories. The classification is evaluated using Precision, Recall, and Matthews Correlation Coefficient (MCC). We conducted an online survey among students in Germany to study their opinions on learning from Stack Overflow, which was analyzed using the open card sort technique. According to our results, React.js is the most in-demand JavaScript framework in 2019, followed by Angular and Vue.js, and Stack Overflow does not directly promote learning for the students. Still, a small fraction of all the data generated on Stack Overflow does. The results also indicate that learning on Stack Overflow is beneficial but comes with some challenges, such as inefficient answers.

Keywords: Stack Overflow, JavaScript Frameworks, Learnability, Students, Germany

Contents

Contents	iv
List of Figures	vi
List of Tables	viii
List of Abbreviations	x
1 Introduction	1
1.1 Motivation	3
1.2 Research Objectives & Questions	4
1.3 Thesis Outline	5
1.4 Considerations	5
2 State of Art	6
2.1 Categorizing Stack Overflow Posts	6
2.2 JavaScript Frameworks	8
2.3 The Proposed Idea	9
3 Methodology	10
3.1 Popular JavaScript Frameworks	10
3.1.1 Selecting Frameworks	10
3.1.2 Stack Overflow Trends	11
3.1.3 Node Package Manager	12
3.1.4 Google Trends	13
3.2 Learning Factor on Stack Overflow	14
3.2.1 Stack Overflow Q&A Model	14
3.2.2 Stack Overflow Data Dump	15
3.2.3 Selecting the Data	16
3.2.4 Calculating Performance Metrics	29
3.3 University Students in Germany on Stack Overflow	31
3.3.1 Stack Overflow Annual Developer Survey	31
3.3.2 Selecting Questions	33
3.3.3 Filtering Student Responses	33
3.3.4 Analyzing Student Responses	34
3.3.5 Survey at Chemnitz University of Technology	43
3.3.6 Survey Planning	43

CONTENTS

3.3.7	Survey Execution	44
3.3.8	Survey Analysis Procedure	44
4	Results	45
4.1	Popular JavaScript Frameworks	45
4.1.1	Stack Overflow Trends	45
4.1.2	Node Package Manager	45
4.1.3	Google Trends	46
4.1.4	Answering RQ-1	47
4.2	Learning Factor on Stack Overflow	47
4.2.1	Categorizing the Data	47
4.2.2	Performance Metrics	48
4.2.3	Answering RQ-2	49
4.3	University Students in Germany on Stack Overflow	49
4.3.1	Stack Overflow Annual Developer Survey Results	49
4.3.2	Student's Survey Result	55
4.3.3	Answering RQ-3	61
5	Discussion	62
5.1	Interpretation of Results	62
6	Threats to Validity	65
7	Conclusion	66
8	Acknowledgments	68
	Bibliography	69

List of Figures

1.1	Usage statistics of JavaScript [1]	1
1.2	Stack Overflow Statistics [4]	3
1.3	Most Popular Technologies [5]	4
3.1	Most Wanted Web Frameworks [5]	10
3.2	Stack Overflow Trends Homepage [22]	11
3.3	Stack Overflow Trends Implementation [22]	12
3.4	NPM Trends Homepage [24]	12
3.5	NPM Trends Implementation [24]	13
3.6	Google Trends Homepage [25]	13
3.7	Google Trends Implementation [25]	13
3.8	Asking a Question [27]	14
3.9	Answer on SO [27]	15
3.10	Stack Exchange Data Explorer New Query [36]	16
3.11	Result of Query from Listing 3.3	20
3.12	Manual Classification	20
3.13	Confusion in manual classification	21
3.14	Classifier Structure	23
4.1	Stack Overflow Trends Results [45]	45
4.2	NPM Trends Results [46]	46
4.3	Google Trends Results [47]	46
4.4	Student Status	49
4.5	Student Age Group	50
4.6	Student Education Level	50
4.7	Student Undergraduate Major	51
4.8	Programming Languages used by Students	51
4.9	Programming Languages desired by Students	52
4.10	Web Frameworks used by Students	53
4.11	Web Frameworks desired by Students	53
4.12	Why Students visited Stack Overflow	54
4.13	Opinions on new content types	54
4.14	Participant's Background	55
4.15	Web Development Experience	55
4.16	Interest/Experience in JavaScript frameworks	56
4.17	Stack Overflow usage	56

LIST OF FIGURES

4.18 Helpfulness on Stack Overflow	57
4.19 Learning with Stack Overflow	57
4.20 Rating the Learning Experience	58
4.21 Challenges faced on Stack Overflow	58
4.22 Stack Overflow: Learning Tool?	59
4.23 Improvements on Stack Overflow	60
4.24 Rank Learning Methodology	60

List of Tables

2.1	Question Categories [6]	7
2.2	Comparison of the learning curve of JavaScript frameworks	9
3.1	Potential metrics for selecting SO posts	16
3.2	Stack Overflow Tags	17
3.3	SEDE Stack Overflow Database Schema [36]	17
3.4	Manual Classification Keywords - Tag: angular	22
3.5	Stack Overflow Survey Results Schema	32
4.1	Manual Classification Results - Top 500 Posts	47
4.2	Automated Classification Results - Random 150 Posts	48
4.3	Performance Metrics - Automated Classification	48

Listings

3.1	Average of Question Score	18
3.2	Average of Answer Count	18
3.3	SQL Query for random 500 posts	19
3.4	post.py - Post model	23
3.5	result.py - Result model	23
3.6	classifier.py - Abstract classifier	24
3.7	review.py - Review classifier	25
3.8	properties.txt	26
3.9	classifier main.py - Main classifier	26
3.10	Loading Libraries and Selecting Questions	33
3.11	Filtering Student Responses	33
3.12	Students From Germany	34
3.13	Student's Age Group	35
3.14	Student Education Level	36
3.15	Student Undergraduate Major	36
3.16	Programming Languages used by Students	37
3.17	Programming Languages desired by Students	38
3.18	Web Frameworks used by Students	39
3.19	Web Frameworks desired by Students	40
3.20	Why Students visited Stack Overflow	40
3.21	What Students want on Stack Overflow	42

List of Abbreviations

HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
Q&A	Question and Answer
SO	Stack Overflow
API	Application Programming Interface
JSF	JavaScript Frameworks
RQ	Research Question
regex	Regular Expression
RF	Random Forest
SVM	Support Vector Machine
NPM	Node Package Manager
SQL	Structured Query Language
CSV	Comma-Separated Values
SEDE	Stack Exchange Data Explorer
IDE	Integrated Development Environment
MCC	Matthews Correlation Coefficient

1 Introduction

A skill that is beneficial in life, in general, is the ability to solve problems. We want to become outstanding problem solvers so that we can solve any difficulty we come across. Learning to program gives us the opportunity to learn this form of skill, and it can enhance us in life along the way. Programming is a useful skill to possess when looking at how the world is developing with smart devices introduced in almost every human life phase, from a watch to an automobile. Programming is the foundation that allows us to use and communicate with computers. In simpler terms, programming can solve real-world problems by interacting with computers in a language it understands. With so many programming languages available today, one such programming language is JavaScript. JavaScript, JS has been adapted as a de facto standard for the World Wide Web. JavaScript proved to be a revolution in the development of web pages at client-side programming. JavaScript changed the way a user interacts with the web pages, such that the user can manipulate different elements of the page and make them more dynamic, including scrolling, printing time and date, creating a calendar, and other tasks that are not possible using plain HTML. As of July 2020, out of all the websites on the internet, 96.6 % are using JavaScript as their client-side programming language [\[1\]](#).



Figure 1.1: Usage statistics of JavaScript [\[1\]](#)

1 Introduction

The developer community adopted JavaScript rapidly at client-side programming, resulting in the birth of various JavaScript libraries and frameworks. One of the earliest JS libraries was JQuery in 2006, but as JS is evolving, many new libraries and frameworks pop up every year. Along with JQuery, some of the libraries and frameworks are Angular, React.js, Vue.js, Ember.js, Meteor, Backbone.js, and many more. As the developer community grows along with various technologies, it is common to face new errors and problems while programming. In 2008, Joel Spolsky and Jeff Atwood decided to do something about it. They built a programming Q&A site that is free. Free to ask questions, free to answer questions, free to read, free to index [2]. This Q&A website's name "Stack Overflow" was chosen in a poll on Jeff Atwood's blog "Coding Horror" [3].

Stack Overflow has become a popular online community where users can share information and learn to code based on the Question and Answer (Q&A) model about various technologies related to programming. Stack Overflow has become a valuable platform to share knowledge among other developers. Another reason to share knowledge on Stack Overflow is no proper documentation or unexpected behavior of a specific technology or API. Users can connect with peers who are working on the same technology to gain and share knowledge.

While providing a common platform for developers, Stack Overflow has gained a massive user base of 13 million registered users with 9.9 million visits per day [4]. Approximately 20 Million questions have been asked on Stack Overflow spread across various programming languages [4]. A healthy and active community like Stack Overflow is asking around 7.4k questions daily [4]. The Stack Overflow community is generating a massive amount of data with highly knowledgeable value daily. Such data is only categorized based on programming languages. This data has the potential to be a knowledge hub for community members who are willing to improve their programming skills or learn a new programming language. To evaluate the potential of the data on Stack Overflow, we analyze the questions and answers on Stack Overflow by classifying them into different question categories. This classification will help us understand the data, promoting useful learning, or not for its community members. As there are many questions with various programming languages, we narrow it down to only JavaScript questions due to its popularity in client-side programming. JavaScript, having a massive ecosystem of libraries and frameworks, we consider the most wanted top 3 JavaScript frameworks by the Stack Overflow community mentioned in SO annual developer survey 2019 results. Initially, the thesis focuses on studying the most trending JS libraries and frameworks from these top 3 JS libraries and frameworks from 2016 to 2019 from various data-driven sources. Further, It explores these framework's learnability by classifying their questions asked on Stack Overflow and evaluating the classification with multiple performance metrics. Finally, we analyze student's experiences from Germany, especially from the Chemnitz University of Technology, while learning JS frameworks with Stack Overflow.

1.1 Motivation

Stack Overflow has evolved from just a Q&A community to a knowledge hub for various types of developers ranging from new to expert developers. As of July 2020, Stack Overflow (SO) has around 13 million registered users and around 20 million questions until today, and the community provides around 30 million supportive answers. Out of all these questions, about 70% have an answer. Stack Overflow is visited by around 9.9 million users every day [4]. Researchers can gain insights into different technologies from such a large programming community.

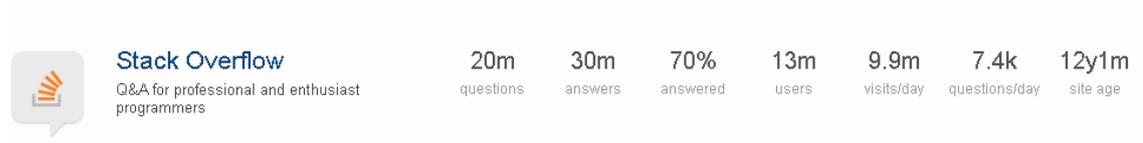


Figure 1.2: Stack Overflow Statistics [4]

In 2011, Stack Overflow started conducting an annual developer survey on its platform. The annual developer survey is conducted worldwide and would gather information about the developer's favorite technologies, environments, job preferences, salaries, and many more aspects concerning the programming community. At the end of each year, Stack Overflow publishes the results of the annual developer survey. These results may not represent the entire developer community but can be considered a glimpse of what is going on worldwide. The last annual developer survey available is for the year 2019. The annual developer survey 2019 received over 88,000 responses from 179 countries and dependent territories [5].

According to the annual developer survey 2019 results, JavaScript is the most commonly used programming language for the seventh year in a row [5]. These results lead to a discussion about how JavaScript is the most used language in the developer community, with a share of 67.8 % [5]. A survey result on an online community cannot be accepted unless a piece of evidence supports it. These results create various opportunities for the scientific community to question and research the online developer community's analysis.

Programming, Scripting, and Markup Languages

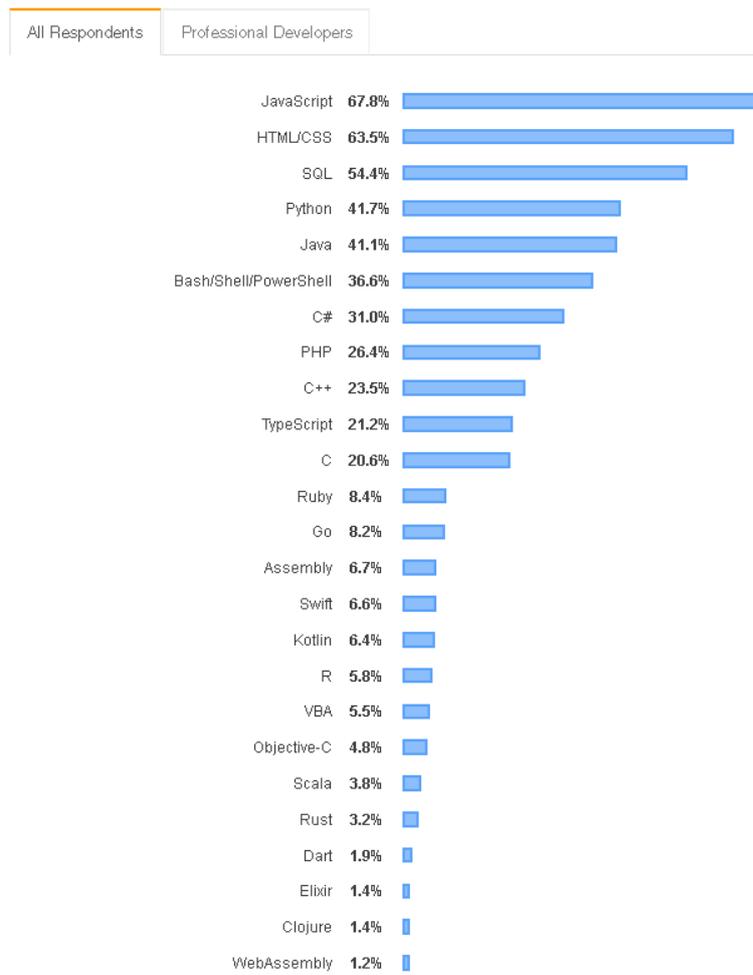


Figure 1.3: Most Popular Technologies [5]

1.2 Research Objectives & Questions

Stack Overflow’s annual developer survey result provides many interesting insights about different technologies around the world. The results concerning specific technologies or other factors argue which one is the most used or least used. The research objective is to study these results to evaluate how such a vast programming community supports learning. Another objective is to identify what type of users are using Stack Overflow. Users can be an expert developer or just a university student.

The thesis evaluates the following research questions to investigate the research objectives.

- **RQ-1:** What are the most in-demand JavaScript frameworks in 2019?
- **RQ-2:** How is Stack Overflow helping developers learn JavaScript frameworks?
- **RQ-3:** How are university students in Germany learning JavaScript frameworks on Stack Overflow?

1.3 Thesis Outline

- Chapter 2, State of Art, beginning page 6, lays the foundation for work and expertise in the specific research area.
- Chapter 3, Methodology, beginning page 10, addresses information on the application of approaches and techniques. This section provides a complete procedure about how the research is designed and performed.
- Chapter 4, Results, beginning page 45, displays the implementation results and analyzes its importance and consequences.
- Chapter 5, Discussion, beginning page 62, explores how good the thesis has been, both in analytical findings and in general.
- Chapter 6, Threats to Validity, beginning page 65, explore our result's internal and external threats.
- Chapter 7, Conclusion, beginning page 66, summarizes the purpose, arguments, and thesis results.

1.4 Considerations

In the thesis, JavaScript frameworks or libraries, irrespective of their type, are referred to as a “framework.” For example, React.js is a library but referred to as a framework to avoid confusion. In the case of Angular, this thesis conforms strictly to Angular version 2 and above, also known simply as “Angular.” Angular versions < 2, also known as “AngularJS,” are beyond the reach of this thesis as we believe that for an upcoming development project, you are testing new, stable technologies.

2 State of Art

This chapter presents a literature overview of related work concerning the learning curve of various JavaScript Frameworks (JSF). This chapter also presents a literature overview of various research that categorized Stack Overflow posts. Further, we discuss the proposed idea. This chapter is divided into three sections:

- The **first section** discusses various categories in which Stack Overflow posts can be categorized.
- The **second section** discusses the various research concerning the learnability of JavaScript frameworks.
- The **third section** presents the proposed idea for the thesis.

2.1 Categorizing Stack Overflow Posts

The data generated by the developer community on Stack Overflow is available for public use. A research paper “Automatically Classifying Posts into Question Categories on Stack Overflow” published in “International Conference on Program Comprehension 2018” by S. Beyer, C. Macho, M. Pinzger, and M. Di Penta [6] suggests classifying SO data into question categories. It compares the taxonomies found by previous researchers such as Christoffer Rosen and Emad Shihab [7], Miltiadis Allamanis and Charles Sutton [8], Christoph Treude, Ohad Barzilay, and Margaret-Anne Storey [9], Stefanie Beyer and Martin Pinzger [10], Stefanie Beyer, Christian Macho, Massimiliano Di Penta, and Martin Pinzger [11]. These taxonomies are harmonized into seven-question categories mentioned in Table 2.1. It also states that these question categories do not vary across programming languages [6].

The research paper proposes manual classification and automated classification with machine-learning algorithms like Random Forest (RF) and Support Vector Machine (SVM) to classify Stack Overflow posts tagged with “Android” into the mentioned question categories. These classifications were compared based on their performance metrics, and RF performed way better than SVM [6]. A research paper, “What kind of questions do developers ask on Stack Overflow? A comparison of automated approaches to classify posts into question categories” [12] further extends this research, which compares automated approaches to classification with a regex approach and regex approach outperformed all the machine-learning approaches with an average precision, recall, and MCC of 0.90, 0.90, and 0.68. The extended

research concludes with the regex approach being the most efficient way to classify Stack Overflow posts [12]. The thesis considers the “Learning” and “Review” category to be useful as learning content on Stack Overflow. The thesis mainly concentrates on studying the learning content available on Stack Overflow.

Categories	Description
API Change	These posts contain questions that arise due to the changes in an API or due to compatibility issues between different versions of an API.
API Usage	The posts falling into this category contain questions asking for suggestions on how to implement some functionality or how to use an API. The questioner is asking for concrete instructions.
Conceptual	The posts of this category consist of questions about the limitations of an API and API behavior, as well as about understanding concepts, such as design patterns or architectural styles, and background information about some API functionality
Discrepancy	The posts of this category contain questions about problems and unexpected behavior of code snippets whereas the questioner has no clue how to solve it.
Learning	In these posts, the questioners ask for documentation or tutorials to learn a tool or language. In contrast to the first category, they do not aim at asking for a solution or instructions on how to do something. Instead, they aim at asking for support to learn on their own.
Errors	Posts of this category deal with problems of exceptions and errors. Often, the questioner posts an exception and the stack trace and asks for help in fixing an error or understanding what the exception means
Review	Questioners of these posts ask for better solutions or reviewing their code snippets. Often, they also ask for best practice approaches or ask for help to make decisions, for instance, which API to select

Table 2.1: Question Categories [6]

2.2 JavaScript Frameworks

“JavaScript UI frameworks and libraries work in cycles. Every six months or so, a new one pops up, claiming that it has revolutionized UI development. Thousands of developers adopt it into their new projects, blog posts are written, Stack Overflow questions are asked and answered, and then a newer (and even more revolutionary) framework pops up to usurp the throne.” [13] (Ian Allen, 2018)

For modern front-end web development, JavaScript frameworks provide developers with tried and tested tools for building scalable, interactive web applications. JavaScript frameworks have reduced the time required to develop a web application compared to just JavaScript [14]. As a standard part of their tooling, many modern firms use frameworks; so many front-end development jobs now require framework experience [15]. There are so many different frameworks available, while new ones emerge all the time [15]. It is not easy to figure out where to start when studying frameworks as an aspiring front-end developer. Amantia Pano, Daniel Graziotin, Pekka Abrahamsson [16] reported that learnability, the developer’s effort to learn the framework, is essential for adopting a JavaScript framework.

We discuss various existing research that analyzes the learnability of JS frameworks against multiple benchmarks. It would help us understand the JS framework’s learning curve as it might directly affect the frequency of questions asked on Stack Overflow and even the usage pattern of these frameworks.

Brandon Satrom [17], Eric Wohlgethan [18], Elar Saks [19], Nicklas Ockelberg and Niclas Olsson [20], compared their experience while learning these JavaScript frameworks with the help of the official technical documentation available on their respective websites. The learning of these JS frameworks can be subjective to the individual researcher, and it can’t be measured quantitatively:

- **Case Study 1:**

Brandon Satrom [17] reported his experience on how quickly he can start using a JavaScript framework and understand the core concepts. The frameworks considered are Angular, React.js, and Vue.js. The idea was to participate in the official tutorial available on these framework’s official website to build a simple demo web application. Brandon Satrom [17] reported that the easiest framework to get started within the shortest amount of time is Vue.js, followed by React.js and Angular.

- **Case Study 2:**

Eric Wohlgethan [18] reported the comparison of the learning curve of Angular, React.js, and Vue.js based on their technical documentation to understand the core concepts and the prerequisites required before learning these frameworks. Vue.js was considered the easiest as it does not require learning any additional language other than JavaScript. However, React.js requires developers to learn

JSX (JavaScript XML) similar to JavaScript, and Angular requires developers to learn TypeScript identical to Java.

- **Case Study 3:**

Elar Saks [19] reported the comparison of the learning curve of Angular, React.js, and Vue.js based on their technical documentation. The learning curve is compared based on learning about the syntax, architecture, data management, lifecycle, and the ease of using third-party libraries of each framework. Vue.js was considered the easiest for a new developer, followed by React.js and Angular. Nicklas Ockelberg and Niclas Olsson [20] carried out similar research on these framework’s usability, and they reported that Vue.js has the smallest learning curve. In contrast, Angular and React.js have an equal learning curve.

Table 2.2 shows a comparison of the learning curve of JavaScript frameworks mentioned in the case studies 1 to 3. As discussed in section 2.1, we concentrate on studying the learning content available on Stack Overflow for JavaScript frameworks; it becomes crucial to study the JavaScript framework’s learning curves. It might directly affect a developer’s question asking pattern on Stack Overflow. For example, suppose a framework is too easy to adapt to in a project. In that case, practitioners of this particular framework will ask fewer questions on Stack Overflow than the practitioners of a complex framework. This question-asking pattern can affect the type of data generated on Stack Overflow. The thesis uses the learning curve of JavaScript frameworks to evaluate their learning content on Stack Overflow.

Authors	Easy	Moderate	Steepest
Brandon Satrom [17]	Vue.js	React.js	Angular
Eric Wohlgethan [18]	Vue.js	React.js	Angular
Elar Saks [19]	Vue.js	React.js	Angular
Nicklas Ockelberg, Niclas Olsson [20]	Vue.js	React.js, Angular	-

Table 2.2: Comparison of the learning curve of JavaScript frameworks

2.3 The Proposed Idea

All these case studies provide the most relevant piece of research for this thesis. Inspired by these various research, the thesis’s proposed idea is to study the learnability content of JavaScript frameworks on Stack Overflow for the developer and the student community.

3 Methodology

3.1 Popular JavaScript Frameworks

- **RQ-1:** What are the most in-demand JavaScript frameworks in 2019?

For answering the RQ-1, we analyze the download statistics of Angular, React.js, and Vue.js using Node Package Manager (NPM) and the frequency of questions asked on Stack Overflow for these frameworks using Stack Overflow Trends. Also, we analyze the popularity of these frameworks on the Google search engine using Google Trends. The most downloaded framework can be considered as the most in-demand framework; However, we also consider the popularity of these frameworks on the Google search engine to evaluate the most downloaded framework. The frequency of questions on Stack Overflow can be the deciding factor for a particular framework's difficulty to learn.

3.1.1 Selecting Frameworks

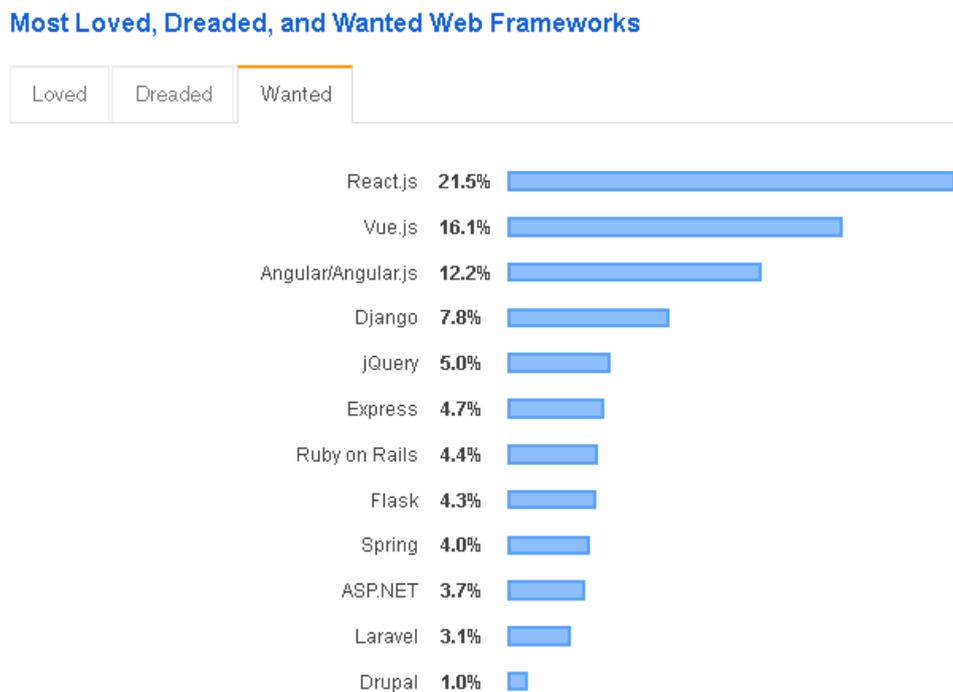


Figure 3.1: Most Wanted Web Frameworks [5]

Figure 3.1 shows the most wanted web frameworks by the developer community on Stack Overflow. Out of 12 web frameworks mentioned, 5 are JavaScript frameworks.

React.js, Vue.js, and Angular/Angular.js are the most wanted web frameworks according to Stack Overflow annual developer survey 2019, as shown in figure 3.1. For the proposed hypothesis, React.js, Vue.js, and Angular are used to answer the research questions.

3.1.2 Stack Overflow Trends

Around 7.4k questions are asked daily on Stack Overflow, as shown in Figure 1.2. These questions can be about programming problems concerning different technologies. In 2017, Stack Overflow developed a new tool called Stack Overflow Trends to track interest in programming languages and technologies, based on the number of Stack Overflow questions asked per month [21]. Any user can check these changing trends over a period of time. This insight cannot be perfect based on questions asked, but it indicates what technologies and programming languages developers community are using and learning [21]. These trends can also be studied based on the level of complexity of different technologies.

Stack Overflow Trends

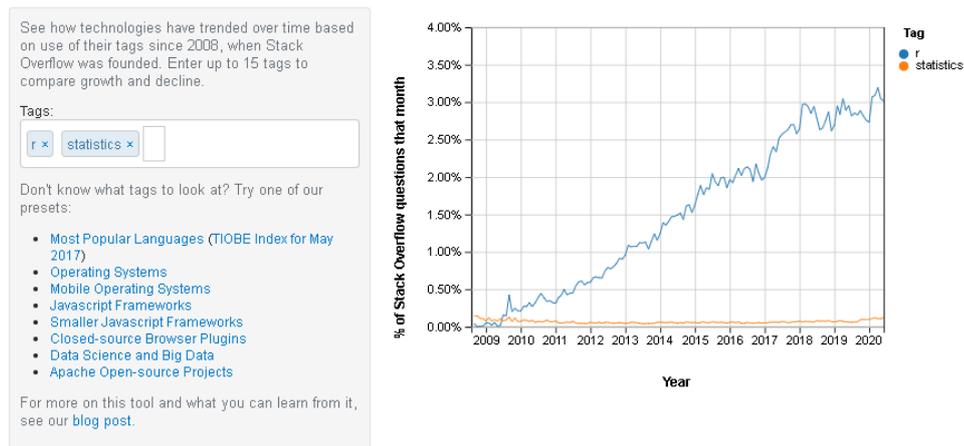


Figure 3.2: Stack Overflow Trends Homepage [22]

As shown in figure 3.2, tags are used to collect the frequency of questions asked on Stack Overflow. Stack Overflow uses tags to link a question to a specific subject area. The frequency of questions is represented on the right side in the form of a graph. The X-axis represents the year in which the questions were asked. The Y-axis represents the percentage of questions with that specific tag from all the questions on Stack Overflow.

To find out the most in-demand framework, the Stack Overflow Trends tool is used to overview the frequency of questions based on specific tags of the selected

frameworks, i.e., React.js, Vue.js, and Angular as shown in figure 3.3.



Figure 3.3: Stack Overflow Trends Implementation [22]

3.1.3 Node Package Manager

NPM, also known as “Node Package Manager,” is an extensive public database of JavaScript software and the meta-information surrounding it [23]. NPM is a public registry that allows JavaScript developers to share their code about solving a particular problem, and other developers can reuse it according to their needs. NPM consists of packages that have JavaScript code in it, along with a meta-information about it. NPM is one of the ways people access various JavaScript packages and frameworks.

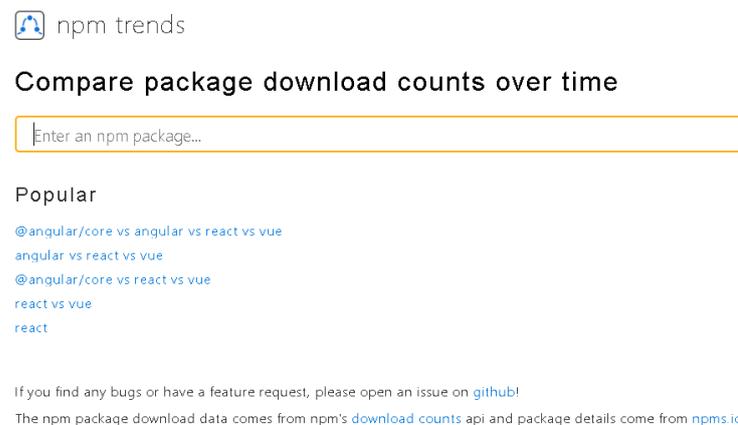


Figure 3.4: NPM Trends Homepage [24]

NPM provides a public API to access download counts of any package on the NPM database. John Potter, the co-founder of Rye Labs, has developed an NPM package comparison application called “NPM Trends,” which can be visited at <https://www.npmtrends.com/> that uses NPM’s public API to represent the download counts of JavaScript packages graphically as shown in figure 3.4. NPM Trends is used to study

the download statistics of the selected frameworks i.e. "angular/core", "react", and "vue" as shown in figure 3.5.



Figure 3.5: NPM Trends Implementation [24]

3.1.4 Google Trends

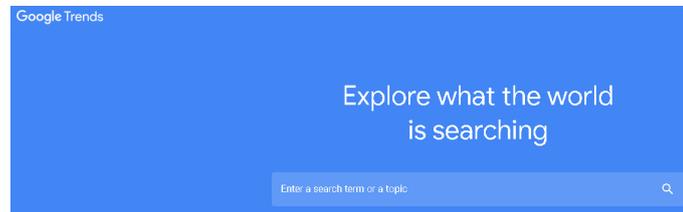


Figure 3.6: Google Trends Homepage [25]

Google Trends is an analysis tool by Google that helps users discover search-behavior or keywords used on Google's search engine. Google search engine is one the most used search engines with 91.89 % market share as of May 2020 [26]. With such a vast market share of web searches, Google Trends lets you see what searches are performed worldwide on its search engine. It provides a volume of keywords searched over a while. It also allows a comparison of the volume of various keywords used in various geographical locations. Keywords like angular, react, and vue is used to get an insight into search trends around the world between 2016-2019 as shown in figure 3.7

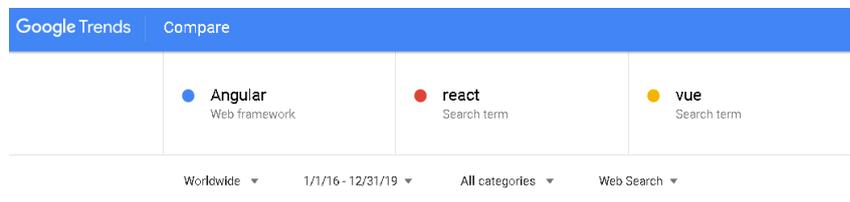


Figure 3.7: Google Trends Implementation [25]

3.2 Learning Factor on Stack Overflow

This section explains the methods required to answer the second research question.

- **RQ-2:** How is Stack Overflow helping developers learn JavaScript frameworks?

RQ-2 concentrates on studying the type of data on Stack Overflow, categorizing the data as learnable or not for the developer community. The methodology dives deep into the content on Stack Overflow and searches for the learnability factor.

3.2.1 Stack Overflow Q&A Model

Stack Overflow has a simple process model. A registered user is allowed to post a question concerning programming. The question format allows the user to enter a “Title” and a “Description” and code snippets supporting the question, as shown in Figure 3.8. The question also allows the user to add tags to the question. Tags can be helpful to relate the question to a subject or a technology.

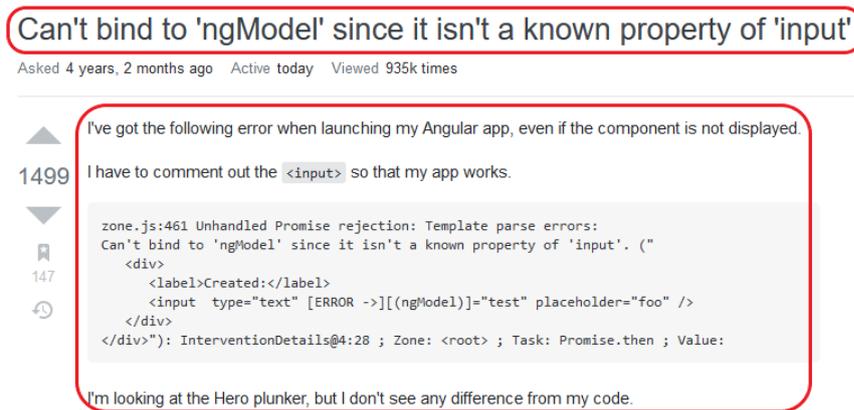


Figure 3.8: Asking a Question [27]

These questions are published in the community. Other registered users can provide answers to these questions. The community can vote on these answers. Answers can be upvoted or downvoted. Good answers are voted up and rise to the top [28]. The asker can mark a question as “accepted,” indicating which answer was the most useful for the asker, as shown in Figure 3.9.

Stack Overflow has introduced gamification in the community with the help of these upvotes and downvotes. A user earns a reputation if the community upvotes his posts (questions, answers, edits). As the user goes on earning reputations, the user unlocks new privileges in the community.

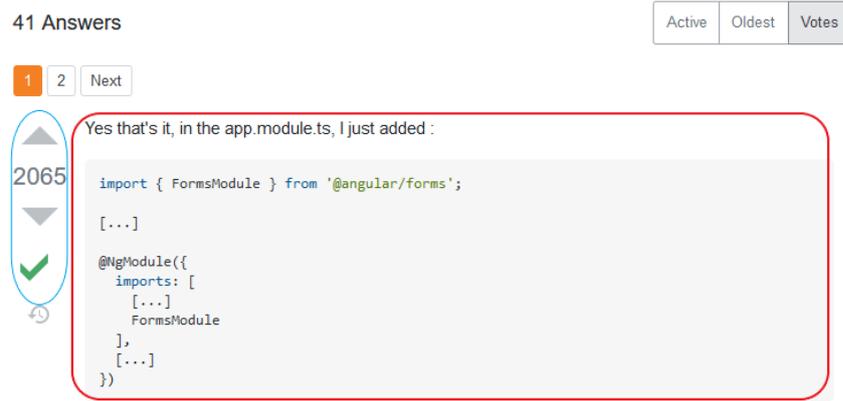


Figure 3.9: Answer on SO [27]

3.2.2 Stack Overflow Data Dump

Stack Overflow decided to make all user-generated data publicly available. All user-generated content on Stack Overflow is under a Creative Commons License [29]. Creative Commons License lets others copy, distribute, display, and perform your copyrighted work and derivative works based upon it, but only if they give credit the way you request [30]. This data should only be used for non-commercial purposes only [30]. Most of the Stack Overflow data is available for the public, which can be used for research purposes. Stack Overflow provides a database dump of anonymized public data every quarter [31].

Such a database dump can be enormous and tedious to handle. To overcome this, Stack Overflow launched a web tool “Stack Exchange Data Explorer” for sharing, querying, and analyzing the database dump of Stack Overflow [32]. Anyone can run a SQL query against this database dumps online and gets a CSV downloadable output, as shown in figure 3.10. The data on Stack Exchange Data Explorer (SEDE) is updated early every Sunday morning around 3:00 UTC [33]. Stack Exchange Data Explorer has a limitation of a maximum of 50,000 rows per query.

Another way to access Stack Overflow data without limitations is Google’s BigQuery. BigQuery lets you run a query without any limitations. It also provides an API to connect all sorts of tools [34]. Google BigQuery is only advantageous over SEDE to query more than 50,000 rows per query and the ability to connect to third-party tools like Tableau [35]. We will be querying a smaller data set than 50,000 rows for the Stack Overflow data dump classification for this thesis. Therefore SEDE is used to query the Stack Overflow data.

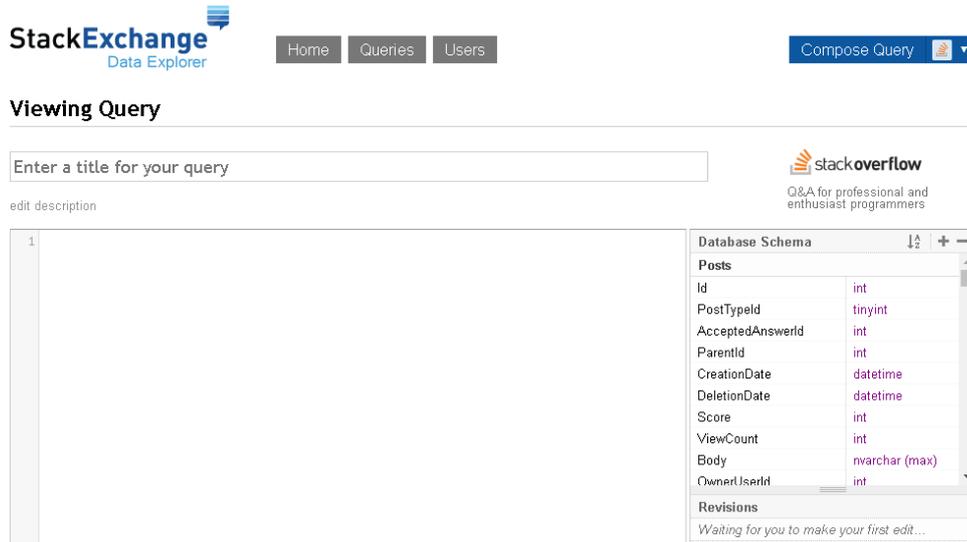


Figure 3.10: Stack Exchange Data Explorer New Query [36]

3.2.3 Selecting the Data

Stack Exchange Data Explorer (SEDE) provides a database schema to query the Stack Overflow Data Dump, shown on the right side of figure 3.10. The initial step is to select top quality posts for manual classification to find the ground truth, which can test the automated classification approach. A research paper, “Predicting the Quality of Questions on Stack Overflow,” published in “Proceedings of Recent Advances in Natural Language Processing, pages 32-40, Hissar, Bulgaria, 2015,” discusses various aspects of predicting the quality of questions. It concludes that the higher the reputation of Question Score and Answer Counts, the higher is the value of the quality measure. Based on this study, the thesis proposes to use Question Score and Answer Counts of the posts to form the ground truth. Since the value of the quality measure is higher when reputation is higher, metrics of Question Score and Answer Counts is set to be greater than their respective average, as shown in Listing 3.3. Table 3.1 shows the potential metrics used by the thesis for selecting the data.

Metrics	Description
Accepted Answer	The asker can mark a question as “accepted,” indicating which answer was the most useful for the asker.
Question Score	The score is calculated by subtracting the down-votes from up-votes
Answer Counts	The number of answers a question has received.

Table 3.1: Potential metrics for selecting SO posts

3 Methodology

Tags
angular
reactjs
vue.js

Table 3.2: Stack Overflow Tags

Table 3.2 shows Stack Overflow tags used to query data on Stack Exchange Data Explorer (SEDE). A user from the Stack Overflow community has the freedom to create his tags after he has earned a specific reputation and unlocked a distinct privilege within the gamification framework of Stack Overflow. There can be multiple tags for the same programming language. That is why we mention the tags used for this thesis in the Table 3.2. Table 3.3 shows the Stack Exchange’s Stack Overflow Database Schema, which the thesis uses. The Stack Overflow database schema helps to understand the structure and datatype in the database. As mentioned in Table 3.3, “Title” and “Body” has a datatype of “nvarchar,” which indicates that these fields have string content whereas “Score,” “AnswerCount,” and “ViewCount” has a datatype of “int” showing only an integer will be retrieved for these fields. The database schema gives the exact information on what type of data to expect when querying a specific field.

Posts	
Id	int
Title	nvarchar (250)
Body	nvarchar (max)
Score	int
AnswerCount	int
ViewCount	int
CommentCount	int
AcceptedAnswerId	int
CreationDate	datetime
PostTags	
PostId	int
Tags	
Id	int

Table 3.3: SEDE Stack Overflow Database Schema [36]

Querying the Data

The initial step is to retrieve the average of Question Score and Answer Counts, as shown in Listing 3.1 and Listing 3.2. We use “AVG(),” an inbuilt function of SQL which calculates the average from the dataset provided to it. We use the “Score” and “AnswerCount” fields from the table “Posts” in the “AVG()” function as shown in Listing 3.1 and Listing 3.2. We perform a “JOIN” function between table “Posts,” “PostTags,” and “Tags” to combine the tags associated with their specific posts. We use a “WHERE” condition to filter the posts based on the period of the year, tag, and an accepted answer. The “TagName” is obtained from the user during runtime as a string on SEDE based on Table 3.2. We use “AcceptedAnswerId” to filter the posts with an accepted answer only by setting it to “NOT NULL” while “CreationDate” is used to set a period between 01 January 2016 and 31 December 2019.

Listing 3.1: Average of Question Score

```
SELECT AVG (p.Score)
FROM Posts p
JOIN PostTags pt ON p.Id = pt.PostId
JOIN Tags t ON pt.TagId = t.Id
JOIN Posts a ON p.AcceptedAnswerId = a.Id
WHERE (t.TagName = ##TagName:string##) AND
(p.AcceptedAnswerId IS NOT NULL) AND
(p.CreationDate >= '##Date1?2016-01-01##') AND
(p.CreationDate <= '##Date2?2019-12-10##')
```

Listing 3.2: Average of Answer Count

```
SELECT AVG (p.AnswerCount)
FROM Posts p
JOIN PostTags pt ON p.Id = pt.PostId
JOIN Tags t ON pt.TagId = t.Id
JOIN Posts a ON p.AcceptedAnswerId = a.Id
WHERE (t.TagName = ##TagName:string##) AND
(p.AcceptedAnswerId IS NOT NULL) AND
(p.CreationDate >= '##Date1?2016-01-01##') AND
(p.CreationDate <= '##Date2?2019-12-10##') AND
(p.Score > 2)
```

The final query is similar to Listing 3.2 but with some modifications. Instead of the “AVG()” function used previously, we use a “TOP(500)” function followed by the fields from the Table 3.3. The “TOP(500)” function would give us the 500 posts from the complete dataset. Further, we add more conditions in the “WHERE” clause. The “Score” and “AnswerCount” is added to the “WHERE” clause with a condition greater than the average of “Score”

3 Methodology

and “AnswerCount” received from Listing 3.1 and Listing 3.2. Lastly, we select these 500 posts randomly from the retrieved dataset by adding the “ORDER BY RAND()” function to our “WHERE” clause. The remaining conditions remain the same as Listing 3.2, as shown in Listing 3.3. A direct link to the query: <https://data.stackexchange.com/stackoverflow/query/1320258/final-query-questions-with-answer-along-with-tag-random-top-500>

Listing 3.3: SQL Query for random 500 posts

```
SELECT TOP(500) p.Id , p.Title , p.Body , p.Score ,  
                p.ViewCount , p.AnswerCount , p.CommentCount  
FROM Posts p  
JOIN PostTags pt ON p.Id = pt.PostId  
JOIN Tags t ON pt.TagId = t.Id  
JOIN Posts a ON p.AcceptedAnswerId = a.Id  
WHERE (t.TagName = ##TagName:string##) AND  
(p.AcceptedAnswerId IS NOT NULL) AND  
(p.CreationDate >= '##Date1?2016-01-01##') AND  
(p.CreationDate <= '##Date2?2019-12-31##') AND  
(p.Score > 2) AND  
(p.AnswerCount >2)  
ORDER BY RAND()
```

Figure 3.11 shows the dataset retrieved from the SQL code from Listing 3.3 for the tag “angular.” The query executes against the database with all the three tags discussed in Table 3.2. These results are downloaded as CSV files and used for manual and automated classifications. “Title” and “Body” gives a stream of data for searching keywords and phrases. “Score” and “AnswerCount” is our potential metrics for filtering the dataset. These metrics for the retrieved dataset are greater than their average. The average is 2 in this dataset of tag “angular.” Hence, this dataset’s minimum score and the minimum number of answers is 3, and the maximum can be any value depending on these question’s random selection. Since the dataset is randomly selected, we can see there is a combination of low score questions and high score questions, and even the “Id” is not in a linear sequence. For the tag “angular,” the maximum score is 1497, and the minimum is 3. The maximum score and minimum score are retrieved using the same query as shown in Listing 3.3 by replacing the parameters after the “SELECT” command with “MAX(p.Score)” and “MIN(p.Score)” respectively, as shown in the query: <https://data.stackexchange.com/stackoverflow/query/1320273/max-score>

3 Methodology

Results		Messages					Download CSV
Id ^	Title ^	Body ^	Score ^	ViewCount ^	AnswerCount ^	CommentCount ^	
35975879	angular2 test, how do I mock...	<p>How do I mock sub component...	61	43873	4	4	
34556277	Angular2 table rows as com...	<p>I am experimenting with angula...	103	60949	5	0	
34979260	Angular 2 - Apply conditional...	<p>I am trying to apply a class to ...	7	14784	3	0	
35296704	Angular2 - how to call comp...	<p>I am using a javascript Object t...	72	87796	6	4	
36126837	Redirect to a new page in An...	<p>I have a scenario say a <stron...	8	41886	3	4	
35082657	Angular2 SVG xlink:href	<p>I have component for rendering...	44	19233	3	0	
34742023	How to clear form after subm...	<p>I have some simple angular 2 c...	86	198966	15	0	
36106350	Attribute directive with ngMo...	<p>I want to change (force) input fi...	23	29112	4	0	
36354325	Angular 2 - NgFor using num...	<p>...for example...</p> <pre><co...	205	257854	15	2	
35521583	How to implement Custom A...	<p>1. Is it even ...	26	17519	3	0	
36494938	Debug & Run Angular2 Type...	<p>Debug & Run An...	79	61975	10	0	
36026428	Angular2: Show placeholder ...	<p>Goal: Load ...	43	61177	8	0	
36271314	Observable errors with Angul...	<p>Hello,</p> ...	7	11305	3	0	
35665358	How to set Content-Type and...	<p>How to set Content-T...	26	76128	3	0	
35326689	How to catch exception corr...	<p>Part of my code:</str...	134	241735	5	0	
35525436	Cannot find module 'glob'	<p>Tell me what extra inf...	18	27343	4	0	

500 rows returned in 38 ms

Figure 3.11: Result of Query from Listing 3.3

Manual Classification

Id	Title	Category	Phrases
34556277	Angular2 table rows as component	Api_Usage	How can i
34559212	How to add form validation pattern in Angular 2?	Api_Usage	What is the right property
34561168	Angular exception: Can't bind to 'ngForIn' since it i	Errors	the error is
34561346	Angular2 exception: Token must be defined	Errors	error
34565097	@ContentChildren is not being populated	Review	am i missing something? / does anyone know
34565872	How to delete compiled JS files from previous type	Api_Usage	how to
34568213	Angular 2: Do something with @Input inside Com	Api_Usage	how can i
34568497	How to set dynamic id in *ngFor?	Api_Usage	how to
34569094	What is the Angular equivalent to an AngularJS \$v	API_Change	What is the equivalent
34569387	Angular 2: Inject view/DOM into component constr	Api_Usage	how to
34571517	Angular: How to determine active route with param	Conceptual	Is there any way
34574167	Angular2 - should private variables be accessible i	Api_Usage	should I be able
34578381	Angular 2 NgModel doesn't work	API_Change	my transition from
34582258	How to export typescript enum with systemjs and	Api_Usage	Why can't I
34583073	Angular 2 Errors and Typescript - how to debug?	Api_Usage	how to
34595343	Angular2 fallback route	Api_Usage	How can I
34597835	How to get current route	Api_Usage	how is that possible
34599174	How to handle query parameters in angular 2	Api_Usage	how do I
34607990	Unit testing an observable in Angular 2	Conceptual	What is the correct way
34608361	How to reset form validation on submission of the	Conceptual	is there any method
34608814	Bidirectional data binding on a component input p	Conceptual	Is there a way to make this works simply
34613065	What is the difference between component and dir	Conceptual	what are the differences between
34614743	How to prepare release version with SystemJS an	Errors	here are the errors
34614818	Angular2 - root relative imports	Conceptual	Is there any way how to
34615418	How to get Dom's height in angular2?	Api_Usage	how to
34615425	How to watch for form changes in Angular	Api_Usage	how to
34616248	Oauth 2 popup with Angular 2	API_Change	upgrading
34619782	Coding against Angular 1, for easiest upgrade to A	API_Change	I'd like to easily upgrade
34628536	ASP.NET 5 + Angular 2 routing (template page no	Api_Usage	how to
34629517	File 'app/hero.ts' is not a module error in the cons	Discrepancy	what's wrong
34634530	Accessing element in an Angular2 .JavaScript ES5	Api_Usage	How can I

Figure 3.12: Manual Classification

3 Methodology

Manual classification is carried out with an excel spreadsheet by going through every question title and body and finding keywords or phrases that are classified into question categories mentioned in Table 2.1. Figure 3.12 shows a glimpse of how a manual classification would look like. The result of this manual classification is considered ground truth. Results of manual classification can differ from researcher to researcher as some phrases might belong to more than one question category. Figure 3.13 shows a post with id 34736161 having phrases for question category “Review” and “API Usage.” The phrase highlighted in red indicates that the post belongs to the question category “API Usage.” Still, another phrase highlighted in yellow indicates the post belongs to the question category “Review.” In this case, we considered this post belongs to the question category “Review” because the user is asking for a better way to do the task by seeking a review of an existing code written by the user.

Create a dropdown component

Asked 4 years, 9 months ago Active 7 months ago Viewed 114k times

I want to create a dropdown menu using Angular 2, but I'm not sure how to do it in the "Angular 2 way".

21 I could create a dropdown component that is used like this:

```
<dropdown>
  <li (click)="action('item 1')">Item 1</li>
  <li (click)="action('item 2')">Item 2</li>
</dropdown>
```

This seems nice, but then the `action` method needs to be defined on the component that contains the `<dropdown>` and the `` elements don't get styles applied from the styles in the `<dropdown>` component, which is kind of odd.

Another option is to create components that are used like this:

```
<dropdown>
  <dropdown-item (click)="action('item 1')">Item 1</dropdown-item>
  <dropdown-item (click)="action('item 2')">Item 2</dropdown-item>
</dropdown>
```

This is more verbose, the `dropdown-item` component handles the click action, and the styles of the items get defined by the `dropdown-item` component as well.

Is there a more canonical way to do this in Angular 2?

Edit: I'm not talking about a custom select input for a form. More like a menu with options, or a right click context menu.

Figure 3.13: Confusion in manual classification

Manual classification gives a set of keywords or phrases. These keywords are combined with similar keywords according to their frequency. We convert keywords to a regex pattern for automated classification. Table 3.4 shows a set of frequently

occurring keywords from a data set that was retrieved with an “angular” tag, as shown in figure 3.11.

Manual Classification		
Category	# of posts	most frequently used phrases
API usage	176	-
Conceptual	86	-
Discrepancy	56	-
Errors	38	-
Review	89	canonical/correct/best/easy/simpler/better/easiest/other/easier/smart/proper/clean/simple/only/cleaner/efficient/elegant - way/way of doing, best/recommended - practice, best approach, simpler/better - solution
API change	9	-
Learning	5	guide and explain, where should I get, find a tutorial, any resources

Table 3.4: Manual Classification Keywords - Tag: angular

Automated Classification

This section describes how to set up an experiment to automatically classify posts into question categories using the regex approach. For an independent sample-set, regex is the most efficient than machine learning algorithms like Random Forest (RF) and Support Vector Machine (SVM) for classification with average precision, recall, and MCC of 0.90, 0.90, and 0.68. [12].

The keywords retrieved in manual classification did not cover all the possibilities to indicate a question category. We started stemming the keywords and combining them with different combinations that would lead to a meaningful phrase. With such combinations, we can catch different tenses. For example, for “REVIEW,” we combined the phrases “simple solution” and “simpler solutions” to the pattern (simple(r)?) solution(s)?.

The code for classification is written in Python programming language with Py-Charm IDE. Figure 3.14 shows the layout for our classifier project folder. For the automated classification, we retrieve a dataset of 150 random posts for each tag without any threshold metrics except “AcceptedAnswerId,” which is set as “NOT NULL,” similar to our initial SQL query mentioned in Listing 3.3. We place our random 150 posts CSV file for each tag from the SEDE in the folder “data.”



Figure 3.14: Classifier Structure

Initially, a model is created for “Post” and “Result” in the “model” folder, depicting the structure of our posts and results, as shown in Listing 3.4 and 3.5. The “Post” model represents the structure of data we have retrieved from SEDE, which is “Id,” “Title,” and “Body.” For the “Result” model, we have an “Id” for the post, “a_reason” for the indication of whether a post belongs to a specific category or not, “points” to calculate the match of our keyword patterns in the title and body, and “support” to specify which phrase is matching to our keyword patterns.

Listing 3.4: post.py - Post model

```

class Post(object):

    def __init__(self, id, title, body):
        self.__id = id
        self.__title = title
        self.__body = body
        return;

    @property
    def id(self):
        return self.__id

    @property
    def title(self):
        return self.__title

    @property
    def body(self):
        return self.__body

```

Listing 3.5: result.py - Result model

```

class Result(object):

    def __init__(self, id, a_reason, points, support):
        self.__id = id
        self.__a_reason = a_reason
        self.__points = points
        self.__support = support
        return;

    @property
    def id(self):
        return self.__id

    @property
    def a_reason(self):
        return self.__a_reason

    @property
    def support(self):
        return self.__support

```

Next, an abstract “Classifier” is created to classify posts using regex in the “classifier” folder, as shown in Listing 3.6.

Listing 3.6: classifier.py - Abstract classifier

```

class AbstractClassifier(ABC):

    def __init__(self):
        # self._post = post;
        self.__name = self.__class__.__name__;
        return;

    def countMatches(self, pattern, text):
        p = re.compile(pattern.lower())
        m = p.findall(text)
        return len(m), m

    @abstractmethod
    def classify(self, post):
        pass

```

3 Methodology

We create an individual classifier for the “REVIEW” category in the “classifier” folder, which includes the keyword patterns we created for the regex approach, as shown in Listing 3.7. An individual classifier for the “LEARNING” category is similar to the “REVIEW” category, but it will have its own keyword patterns. This individual classifier calls the abstract classifier (Listing 3.6), passing their respective keyword patterns and the post to perform a regex search. We calculate the length of the matched keyword patterns as points. In the end, if the points are greater than zero, we label the specific post with that individual classifier name and create a new instance of the “Result” model.

Listing 3.7: review.py - Review classifier

```
class Review( AbstractClassifier ):

    def __init__( self ):
        self.pattern = [ "(?i)way of doing",
                        "(?i)(Did i miss|missing something)",
                        "(?i)(do(ing)?|anything|what is|going|
                        something|already)( [\s]+){0,3} wrong ",
                        "(?i)proper( [\s]+){0,3} type(s)?",
                        "(?i)should I configure", ]
        return ;

    @property
    def name( self ):
        return self.__class__.__name__.upper()

    def classify( self , post ):
        id = post.id

        # dlog( id , ": ")
        count_yes = 0;

        support = []
        for pat in self.pattern:
            found, m = self.countMatches
            (pat, post.title + " " + post.body)
            count_yes += found
            if found > 0:
                support.append([found, pat])

        points = count_yes
        if points > 0:
            return Result(post.id, self.name, points,
                support, post.reason);
```

```

else:
    return Result(post.id, "NOT-" + self.name,
                  points, support, post.reason);

```

Listing 3.9 shows the main classifier file from where we call the previous model and functions. Initially, we read properties from a file “properties.txt,” as shown in Listing 3.8. This properties file specifies where our dataset file is stored and delimiter to be used and a place where result files can be saved (“res” folder), as shown in Listing 3.8. The main classifier reads the properties from “properties.txt” then loads the dataset from our CSV files. Further, for each classifier, it calls the abstract classifier to match their patterns using regex. In the end, it writes all the results in a new result file inside the “res” folder. We run this main classifier twice of each tag, one time for each question category along with their keyword patterns.

Listing 3.8: properties.txt

```

[ Classification ]
classification = normal

[ CSV ]
delimiter = ,

[ Files ]
postsfilename = data/angular_150.csv
results_file = res/results_

```

Listing 3.9: classifier_main.py - Main classifier

```

class Classifier_Main(object):

    def __init__(self):

        self.classifiers = [Documentation(), Review()]

        # read properties file
        config = configparser.ConfigParser()
        config.read('properties.txt')

        # read properties from file
        self.postsfilename =
        config['Files']['postsfilename']
        self.delimiter = config['CSV']['delimiter']
        self.results_file = config['Files']['results_file']

        # settings for classification
        self.classification =

```

```

config[ 'Classification '][ 'classification ' ]

return ;

def main( self ):

    log( datetime.datetime.now(), ": START")
    # get posts from db or csv
    posts = self._get_posts()

    if self.classification == 'normal':
        self.classify_normal(posts)

    log( datetime.datetime.now(), ": DONE")
    return ;

def _get_posts( self ):
    if os.path.isfile( self.postsfilename ):
        return self._load_posts( self.postsfilename )

def _load_posts( self , filename ):
    log( datetime.datetime.now(), ": load posts from
    csv: ", filename )
    posts = []

    with open( filename , encoding="utf-8") as csvfile :
        next( csvfile )
        readCSV = csv.reader( csvfile ,
        delimiter=self.delimiter )

        for row in readCSV:
            p = Post( row[0] , row[1] , row[2])
            posts.append(p)

    log( datetime.datetime.now(), )
    self.number_posts=len( posts )
    return posts

def classify_normal( self , posts ):
    log( datetime.datetime.now(), ": NORMAL
    classification selected")

    for cc in self.classifiers :

```

3 Methodology

```
log(datetime.datetime.now(), ": START with
classification for ", len(posts), " posts
into ", cc.name)

results = []
results_to_save=[]
progress = 0
for p in posts:

    value1 = cc.classify(p)

    results_to_save.append(value1)
    progress = progress + 1

    if progress % 150 == 0:
        log(datetime.datetime.now(), " ",
            str(progress) + " posts processed")
        self._save_results_to_csv(cc.name,
            results_to_save)
        results.extend(results_to_save)
        results_to_save = []

self._save_results_to_csv(cc.name,
results_to_save)
if len(results)==0:
    results = results_to_save

def _save_results_to_csv(self, name, results):

    pos = list(filter(lambda x: x.a_reason == name,
results))

    neg = list(filter(lambda x: x.a_reason ==
'NOT-' + name, results))

    log(datetime.datetime.now(), " CLASSIFICATION into
", name, ": pos=", len(pos), ", neg=", len(neg))

    file = open(self.results_file+name+".csv", "a+")

    writer = csv.writer(file, delimiter=self.delimiter)

    for res in results:
        writer.writerow([res.id] + [res.a_reason]+
```

```

        [res.support])
    file.close()

    return

Classifier_Main().main();

```

3.2.4 Calculating Performance Metrics

We used Precision, Recall and Matthews Correlation Coefficient (MCC) [37] [38] metrics for measuring and comparing classifier efficiency. These metrics are used to evaluate the results of our automated classification. Notice that we disclose measurements on both sides of the classification: whether a post was appropriately classified as part of a question category (True Positive) and whether a post was accurately classified as not part of a question category (True Negative). We evaluate the results from automated classification manually and label them with “True Positive” or “True Negative” (if a post is correctly classified in that specific category) and “False Positive” or “False Negative” (if a post is not correctly classified in that specific category). These factors are explained further in our performance metrics. The performance metrics help to understand if the classifier worked as expected or not. If the classifier has a higher accuracy rate, it will support our approach to search for learning content on Stack Overflow. We calculate the performance metrics like Precision, Recall, and MCC for our automated classification results for each tag and present the results in Chapter 4.

Precision

Precision is the ratio of correctly classified posts for all posts categorized into the category of question. The values differ from 0 (low accuracy) to 1 (high accuracy) [12]. In a classification process, precision is the sum of true positives (i.e., the number of items that are correctly classified as a positive class) divided by the overall number of items classified as belonging to the incorrect class (i.e., the total of true positives and false positives, which are items falsely classified as belonging to the class). Precision is defined as: [39]

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \quad (3.1)$$

Recall

Recall is the ratio of correctly classified posts for the posts that are actually observed as true instances. Values range from 0 (low recall) to 1 (high recall) [12]. Recall is the sum of true positives divided by the sum of all items that currently belong to the positive class (i.e., the total of true positives and false negatives, which were not identified as the positive class but that should have been). Recall is defined as: [39]

$$\text{Recall} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative (FN)}} \quad (3.2)$$

Matthews Correlation Coefficient (MCC)

Matthews Correlation Coefficient (MCC) tests binary classifier efficiency by considering correctly categorized items, true positives (TP) and true negatives (TN), and the incorrectly categorized items false positives (FP) and false negatives (FN) [12]. It is calculated by measuring the ratio of the product difference of the correctly classified instances (TP · TN) and the product of the misclassified instances (FP · FN) to the product root of the sum of each combination of TP, TN, FP, and FN [12]. The MCC can be determined directly from the confusion matrix using the following formula: [12]

$$\text{MCC} = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (3.3)$$

The values range from -1 (the total difference between the posts marked and the classifier) to 1 (perfect classifier), and a value of 0 means the classifier is as robust as any random guess [12].

3.3 University Students in Germany on Stack Overflow

This section explains the methods required to answer the third research question.

- **RQ-3:** How are university students in Germany learning JavaScript frameworks on Stack Overflow?

RQ-3 focuses on university student's thoughts and opinions in Germany about their learning experience on Stack Overflow and their most preferred technologies. It would help to understand the mindset of the new generation of upcoming developers from Germany.

3.3.1 Stack Overflow Annual Developer Survey

In 2011, Stack Overflow started conducting an annual developer survey on its platform. At the end of every year, Stack Overflow publishes the results of the annual developer survey. These results may not represent the entire developer community but can be considered a glimpse of what is going worldwide. The last annual developer survey available is for the year 2019. The annual developer survey 2019 received over 88,000 responses from 179 countries and dependent territories [5].

The annual developer survey dataset is available for public use <https://insights.stackoverflow.com/survey> where the personal user data is anonymized. The initial overview of this 2019 data reveals some useful insights. 36073 participants are from Europe, which is one of the highest compared to other continents. 5866 participants are from Germany, out of which 1558 are full-time or part-time students. The survey dataset is available in a CSV format with a data schema. We use a few of the questions to analyze what students from Germany desire and their thoughts on Stack Overflow.

Table 3.5 shows the dataset schema of Stack Overflow annual developer survey results.

3 Methodology

Column	QuestionText
Country	In which country do you currently reside?
Student	Are you currently enrolled in a formal, degree-granting college or university program?
Age	What is your age (in years)? If you prefer not to answer, you may leave this question blank.
EdLevel	Which of the following best describes the highest level of formal education that you've completed?
UndergradMajor	What was your main or most important field of study?
LanguageWorkedWith	Which of the following programming, scripting, and markup languages have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the language and want to continue to do so, please check both boxes in that row.)
LanguageDesireNextYear	Which of the following programming, scripting, and markup languages have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the language and want to continue to do so, please check both boxes in that row.)
WebFrameWorkedWith	Which of the following web frameworks have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the framework and want to continue to do so, please check both boxes in that row.)
WebFrameDesireNextYear	Which of the following web frameworks have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the framework and want to continue to do so, please check both boxes in that row.)
SOVisitTo	I visit Stack Overflow to... (check all that apply)
SONewContent	Would you like to see any of the following on Stack Overflow? Check all that apply.

Table 3.5: Stack Overflow Survey Results Schema

These questions are closed questions and give a quantitative data overview.

3.3.2 Selecting Questions

We use the Jupyter Notebook IDE, which shows the live code and data visualizations in the same file. The complete code is written in a single file, “RQ3_SO_survey_analysis.ipynb.” Initially, we import all the required libraries for data analysis and data visualization. We will be using libraries like Pandas [40] and NumPy [41] for data analysis and Matplotlib [42] and Seaborn [43] for data visualization. As shown in Listing 3.10, we load these python libraries and retrieve the required columns from the survey dataset CSV file, as mentioned in Table 3.5. A variable “df” is created, representing two-dimensional tabular data, referred to as “DataFrame.”

Listing 3.10: Loading Libraries and Selecting Questions

```
# Load required libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from collections import Counter
%matplotlib inline

# Load CSV file
df = pd.read_csv('survey_results_public.csv')

#Filter required Columns
df = df[['Country', 'Student', 'Age', 'EdLevel',
        'UndergradMajor', 'LanguageWorkedWith',
        'LanguageDesireNextYear', 'WebFrameWorkedWith',
        'WebFrameDesireNextYear', 'SOVisitTo', 'SONewContent']]

#Set Bar chart Style
plt.style.use("fivethirtyeight")
```

3.3.3 Filtering Student Responses

In Listing 3.11, Using the “Country” and “Student” columns, we extract all the responses from the country ‘Germany’ and further filter it with full-time and part-time student responses to initialize our data frame with student responses from Germany.

Listing 3.11: Filtering Student Responses

```
#Filter Responses From Country 'Germany'
df_germany = df[df.Country == "Germany"]
```

```
#Filter 'Student' Responses From Country 'Germany'

filter_student = df_germany["Student"].isin
                  (["Yes, full-time", "Yes, part-time"])

df_student = df_germany[filter_student]
```

3.3.4 Analyzing Student Responses

Student Status

In Listing 3.12, we replace all the empty answers in the “Student” column with “I prefer not to say” for a better visualization experience. We use the “countplot()” function from the seaborn library to visualize the data for student and non-student responses from the country “Germany.” In the end, we use a “annotate()” function to calculate and display the percentage of each group of answers. The “countplot()” and “annotate()” function is consistent across all other listings.

Listing 3.12: Students From Germany

```
#Group 1 : Germany - Student Status

# Replacing Empty Answers with "I prefer not to say"

filter_replace_nan = (df_germany["Student"].isnull())
df_germany.loc[filter_replace_nan, 'Student'] =
                  "I prefer not to say"

ax = sns.countplot(y="Student", data=df_germany,
                  order = df_germany['Student'].value_counts().index)
ax.set(xlabel="Answer Count", ylabel='Answers')
plt.title("Germany: Student Status")

total = len(df_germany.Student)
for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_width()/total)
    x = p.get_x() + p.get_width() + 0.02
    y = p.get_y() + p.get_height()/2
    ax.annotate(percentage, (x, y))
```

Student Age Group

In Listing 3.13, we remove the empty answers from the column “Age” and loop through each participant’s age, and group it together into different age groups for better visualization. Further, we visualize these age groups.

Listing 3.13: Student’s Age Group

```
#Group 2.1 : Student Age Group

# Remove rows with - NaN
df_student_temp = df_student
#df_student_temp = df_student_temp.dropna(subset=['Age'])

temp_array = []
for age in df_student_temp['Age']:
    if age < 15:
        temp_array.append("Younger than 15 years")
    if age > 14 and age < 21:
        temp_array.append("15 to 20 years")
    if age > 20 and age < 31:
        temp_array.append("21 to 30 years")
    if age > 30 and age < 41:
        temp_array.append("31 to 40 years")
    if age > 40 and age < 51:
        temp_array.append("41 to 50 years")
    if age > 50:
        temp_array.append("Older than 50 years")

df_new = pd.DataFrame(temp_array, columns=['Age_Groups'])
plt.title("Age Groups")

ax = sns.countplot(y="Age_Groups", data=df_new)
ax.set(xlabel="Answer Count", ylabel='Answers')

total = len(df_student.Age)
for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_width()/total)
    x = p.get_x() + p.get_width() + 0.02
    y = p.get_y() + p.get_height()/2
    ax.annotate(percentage, (x, y))
```

Student Education Level

In Listing 3.14, we use the “EdLevel” column to visualize the education level of students who participated in the survey from Germany using the “countplot()” and

“annotate()” function.

Listing 3.14: Student Education Level

```
#Group 2.2 : Student Education Level (EduLevel)

ax = sns.countplot(y="EdLevel", data=df_student,
                  order = df_student['EdLevel'].value_counts().index)
ax.set(xlabel="Answer Count", ylabel='Answers')
plt.title("Education Level of Students")

total = len(df_student.EdLevel)
for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_width()/total)
    x = p.get_x() + p.get_width() + 0.02
    y = p.get_y() + p.get_height()/2
    ax.annotate(percentage, (x, y))
```

Student Undergraduate Major

In Listing 3.15, we use the “UndergradMajor” column to visualize the undergraduate major stream of students who participated in the survey from Germany using the “countplot()” and “annotate()” function.

Listing 3.15: Student Undergraduate Major

```
#Group 2.3 : Student Undergraduate Major (UndergradMajor)
ax = sns.countplot(y="UndergradMajor", data=df_student,
                  order = df_student['UndergradMajor'].value_counts().
                    index)
ax.set(xlabel="Answer Count", ylabel='Answers')
plt.title("Undergraduate Major of Students")

total = len(df_student.UndergradMajor)
for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_width()/total)
    x = p.get_x() + p.get_width() + 0.02
    y = p.get_y() + p.get_height()/2
    ax.annotate(percentage, (x, y))
```

Programming Languages used by Students

In Listing 3.16, we use the “LanguageWorkedWith” column to visualize student’s programming languages. Initially, we remove the empty answers from the column “LanguageWorkedWith.” Each response has multiple answers in this column divided by a semicolon. We loop through these answers and split them by semicolon, and

save them in an array. We use this array to visualize the used programming languages using the “countplot()” and “annotate()” function.

Listing 3.16: Programming Languages used by Students

```
#Group 3.1 LanguageWorkedWith

# Remove rows with - NaN
df_student_temp = df_student
df_student_temp = df_student_temp.dropna(subset=
    [ 'LanguageWorkedWith ' ])

# Split Answers and add it in a new DataFrame
temp_array = []
for visitTO in df_student_temp[ 'LanguageWorkedWith ' ]:
    splited = visitTO.split( ';' )
    for split in splited:
        temp_array.append( split )

df_new = pd.DataFrame( temp_array ,
    columns=[ 'LanguageWorkedWith ' ])
plt.title( "Languagues Used" )
fig = plt.gcf()
fig.set_size_inches( 15, 11 )

ax = sns.countplot( y="LanguageWorkedWith", data=df_new,
    order = df_new[ 'LanguageWorkedWith ' ].value_counts()
    .index )
ax.set( xlabel="Answer Count", ylabel='Answers ' )
fig = plt.gcf()
fig.set_size_inches( 15, 11 )

total = len( df_student.LanguageWorkedWith )
for p in ax.patches:
    percentage = '{:.1f}%'.format( 100 * p.get_width() / total )
    x = p.get_x() + p.get_width() + 0.02
    y = p.get_y() + p.get_height() / 2
    ax.annotate( percentage , ( x, y ) )
```

Programming Languages Desired by Students

In Listing 3.17, we use the “LanguageDesireNextYear” column to visualize the programming languages desired by students. Initially, we remove the empty answers from the column “LanguageDesireNextYear.” Each response has multiple answers in this column divided by a semicolon. We loop through these answers and split

them by semicolon, and save them in an array. We use this array to visualize the desired programming languages using the “countplot()” and “annotate()” function.

Listing 3.17: Programming Languages desired by Students

```
#Group 3.2 LanguageDesireNextYear

# Remove rows with - NaN
df_student_temp = df_student
df_student_temp = df_student_temp.dropna(subset=
    [ 'LanguageDesireNextYear ' ])

# Split Answers and add it in a new DataFrame
temp_array = []
for visitTO in df_student_temp[ 'LanguageDesireNextYear ' ]:
    splited = visitTO.split( ';' )
    for split in splited:
        temp_array.append( split )

df_new = pd.DataFrame( temp_array ,
    columns=[ 'LanguageDesireNextYear ' ])
plt.title( "Languages Desire" )
fig = plt.gcf()
fig.set_size_inches( 15 , 11 )

ax = sns.countplot( y="LanguageDesireNextYear" , data=df_new ,
    order = df_new[ 'LanguageDesireNextYear ' ]
    .value_counts().index )
ax.set( xlabel="Answer Count" , ylabel='Answers ' )

total = len( df_student.LanguageDesireNextYear )
for p in ax.patches:
    percentage = '{:1f}%'.format( 100 * p.get_width() / total )
    x = p.get_x() + p.get_width() + 0.02
    y = p.get_y() + p.get_height() / 2
    ax.annotate( percentage , ( x , y ) )
```

Web Frameworks used by Students

In Listing 3.18, we use the “WebFrameWorkedWith” column to visualize student’s used web frameworks. Initially, we remove the empty answers from the column “WebFrameWorkedWith.” Each response has multiple answers in this column divided by a semicolon. We loop through these answers and split them by semicolon, and save them in an array. We use this array to visualize the used web frameworks using the “countplot()” and “annotate()” function.

Listing 3.18: Web Frameworks used by Students

```

#Group 3.3 WebFrameWorkedWith

# Remove rows with - NaN
df_student_temp = df_student
df_student_temp = df_student_temp.dropna(subset=
    [ 'WebFrameWorkedWith ' ])

# Split Answers and add it in a new DataFrame
temp_array = []
for visitTO in df_student_temp[ 'WebFrameWorkedWith ' ]:
    splited = visitTO.split( ';' )
    for split in splited:
        temp_array.append( split )

df_new = pd.DataFrame( temp_array ,
    columns=[ 'WebFrameWorkedWith ' ])
plt.title( "Web Framewroks Used" )
fig = plt.gcf()
fig.set_size_inches( 15 , 8 )

ax = sns.countplot( y="WebFrameWorkedWith" , data=df_new ,
    order = df_new[ 'WebFrameWorkedWith ' ]
    .value_counts().index )
ax.set( xlabel="Answer Count" , ylabel='Answers ' )

total = len( df_student.WebFrameWorkedWith )
for p in ax.patches:
    percentage = '{:.1f}%'.format( 100 * p.get_width() / total )
    x = p.get_x() + p.get_width() + 0.02
    y = p.get_y() + p.get_height() / 2
    ax.annotate( percentage , ( x , y ) )

```

Web Frameworks desired by Students

In Listing 3.19, we use the “WebFrameDesireNextYear” column to visualize the web frameworks desired by students. Initially, we remove the empty answers from the column “WebFrameDesireNextYear.” Each response has multiple answers in this column divided by a semicolon. We loop through these answers and split them by semicolon, and save them in an array. We use this array to visualize the desired web frameworks using the “countplot()” and “annotate()” function.

Listing 3.19: Web Frameworks desired by Students

```

#Group 3.4 WebFrameDesireNextYear

# Remove rows with - NaN
df_student_temp = df_student
df_student_temp = df_student_temp.dropna(subset=
    [ 'WebFrameDesireNextYear' ])

# Split Answers and add it in a new DataFrame
temp_array = []
for visitTO in df_student_temp[ 'WebFrameDesireNextYear' ]:
    splited = visitTO.split( ';' )
    for split in splited:
        temp_array.append( split )

df_new = pd.DataFrame( temp_array ,
    columns=[ 'WebFrameDesireNextYear' ])
plt.title( "Web Framewroks Desire" )
fig = plt.gcf()
fig.set_size_inches( 15, 8 )

ax = sns.countplot( y="WebFrameDesireNextYear", data=df_new ,
    order = df_new[ 'WebFrameDesireNextYear' ]
    .value_counts().index )
ax.set( xlabel="Answer Count", ylabel='Answers' )

total = len( df_student.WebFrameDesireNextYear )
for p in ax.patches:
    percentage = '{:.1f}%'.format( 100 * p.get_width() / total )
    x = p.get_x() + p.get_width() + 0.02
    y = p.get_y() + p.get_height() / 2
    ax.annotate( percentage, ( x, y ) )

```

Why Students visited Stack Overflow

In Listing 3.20, we use the “SOVisitTo” column to visualize student’s purpose to visit Stack Overflow. Initially, we replace all the empty answers in the “SOVisitTo” column with “No Answer” for a better visualization experience. Each response has multiple answers in this column divided by a semicolon. We loop through these answers and split them by semicolon, and save them in an array. We use this array to visualize the reasons behind student’s visits to SO using the “countplot()” and “annotate()” function.

Listing 3.20: Why Students visited Stack Overflow

```

#Group 4.1 SOVisitTo

# Replace NaN with "No Answer"
df_student_temp = df_student
filter_replace_nan = (df_student['SOVisitTo'].isnull())
df_student_temp.loc[filter_replace_nan, 'SOVisitTo']
                        = "No Answer"

# Split Answers and add it in a new DataFrame
temp_array = []
for visitTO in df_student_temp['SOVisitTo']:
    splited = visitTO.split(';')
    for split in splited:
        temp_array.append(split)

df_new = pd.DataFrame(temp_array, columns=['SOVisitTo'])
plt.title("Why Visit Stack Overflow")

ax = sns.countplot(y="SOVisitTo", data=df_new,
                  order = df_new['SOVisitTo'].value_counts().index)
ax.set(xlabel="Answer Count", ylabel='Answers')

total = len(df_student.SOVisitTo)
for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_width()/total)
    x = p.get_x() + p.get_width() + 0.02
    y = p.get_y() + p.get_height()/2
    ax.annotate(percentage, (x, y))

```

What Students want on Stack Overflow

In Listing 3.21, we use the “SONewContent” column to visualize new content students would like to see on Stack Overflow. Initially, we replace all the empty answers in the “SONewContent” column with “None of the above” for a better visualization experience. Each response has multiple answers in this column divided by a semicolon. We loop through these answers and split them by semicolon, and save them in an array. We use this array to visualize the new content students would like to see on Stack Overflow using the “countplot()” and “annotate()” function.

Listing 3.21: What Students want on Stack Overflow

```

#Group 4.2 SONewContent

# Replace NaN with "No Answer"
df_student_temp = df_student
filter_replace_nan = (df_student['SONewContent'].isnull())
df_student_temp.loc[filter_replace_nan, 'SONewContent'] =
    "None of the above"

# Split Answers and add it in a new DataFrame
temp_array = []
for visitTO in df_student_temp['SONewContent']:
    splited = visitTO.split(';')
    for split in splited:
        temp_array.append(split)

df_new = pd.DataFrame(temp_array, columns=['SONewContent'])
plt.title("New Content on Stack Overflow")

ax = sns.countplot(y="SONewContent", data=df_new,
    order = df_new['SONewContent'].value_counts().index)
ax.set(xlabel="Answer Count", ylabel='Answers')

total = len(df_student.SONewContent)
for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_width()/total)
    x = p.get_x() + p.get_width() + 0.02
    y = p.get_y() + p.get_height()/2
    ax.annotate(percentage, (x, y))

```

3.3.5 Survey at Chemnitz University of Technology

Stack Overflow's annual developer survey only has closed questions. This analysis is not sufficient to answer the research question about the learning factor on Stack Overflow. To answer the research questions more broadly, a survey with open questions is carried out among students from universities in Germany, especially students from the Chemnitz University of Technology.

3.3.6 Survey Planning

Goals

- **Goal 1:** Analyze how Stack Overflow has helped students compared to a different source.
- **Goal 2:** Analyze the challenges faced by students on Stack Overflow while learning a new programming language.
- **Goal 3:** Analyze the improvements in the learning experience suggested by students on Stack Overflow.

Participants

The plan is to let a minimum of 50 participants take the survey. The participants are computer science students studying at a university in Germany, especially at the Chemnitz University of Technology. We do not pay the participants. The purpose of this survey is explained to the participants, and the participant's motivation is to share the results of the research with them once the results are analyzed.

Survey Materials

The survey is divided into the following four sections with their subsequent questions.

1. Participant Profile

- a) Are you a student at the Chemnitz University of Technology?
- b) Do you have any experience working with web development?
- c) Do you have any experience or interest in learning at least one of the Javascript frameworks or libraries? (e.g. Angular, Reactjs, Vuejs, Nodejs etc)
- d) Have you ever used a Q&A platform like Stack Overflow to solve your coding problems?

2. Helpfulness on Stack Overflow

- a) Open Question: While solving a coding problem, how Stack Overflow helped you compared to a different source?

- b) Do you think you can learn a programming language or any framework/library with the help of Stack Overflow?
- c) On a scale of 1 - 5. How much will you rate your learning experience on Stack Overflow?

3. Learning Challenges on Stack Overflow

- a) Open Question: What challenges do you face on Stack Overflow while learning a new language/framework/library?
- b) Do you think Stack Overflow can be a valuable resource/tool for learning?

4. Learning Improvements for Stack Overflow

- a) Open Question: How can Stack Overflow improve its platform to become a Q&A based learning platform?
- b) When learning a programming language or a framework/library, how will you rank your learning methodology? [YouTube, Official Documentation, Q&A Websites, Other Resource]

3.3.7 Survey Execution

Procedure

- The survey is to be carried out online with the help of Google Forms.
- The responses are collected and stored in Google Drive and can be accessed with the help of Google Sheets.
- Participants were sent a link to the survey via different communication channels (WhatsApp, E-Mail.)

Deviations

- The participants can be from different universities other than the Chemnitz University of Technology.
- Participants can be recently graduated students working as a full-time professional.

3.3.8 Survey Analysis Procedure

- Closed questions are represented on a quantitative basis.
- Open questions are evaluated based on Open Card Sorts. Similar data is grouped in Open Card Sort, and these groups are labeled and represented quantitatively. [44].

4 Results

4.1 Popular JavaScript Frameworks

- RQ-1: What are the most in-demand JavaScript frameworks in 2019?

4.1.1 Stack Overflow Trends

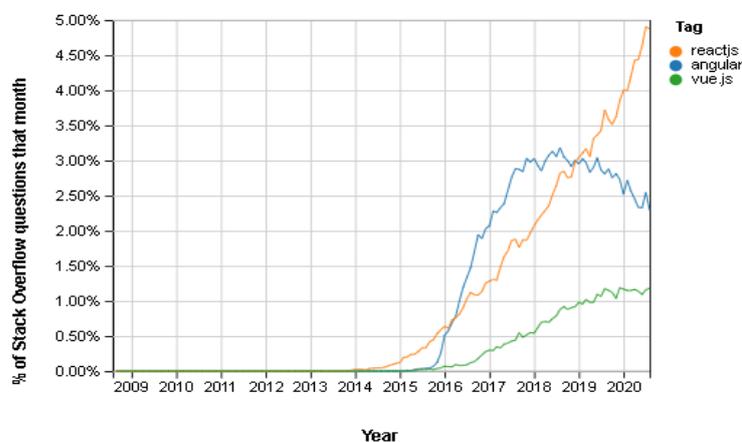


Figure 4.1: Stack Overflow Trends Results [45]

Figure 4.1 shows the percentage of questions of each JavaScript framework asked on Stack Overflow. Angular was the leading JavaScript framework with around 3 % share at its peak of all the questions asked on Stack Overflow from 2016 to 2019. React.js was growing gradually along with Angular since 2014 but crossed Angular in 2019 and now has the highest percentage of questions in 2020 (August) asked on Stack Overflow among JavaScript frameworks with a 5 % share. Vue.js is slowly gaining momentum with around 1 % of questions on Stack Overflow.

4.1.2 Node Package Manager

Figure 4.2 shows the download statistics of our selected JavaScript frameworks from 2016 to 2020. React.js is leading from 2016 to 2020 (August) with around 6 million downloads in 2019 to reaching 8 million downloads in 2020. Angular and Vue.js statistics are similar to a difference of a few thousand downloads. Both are downloaded around 2 million times in 2020 (August).

4 Results

Downloads in past 5 Years ▾

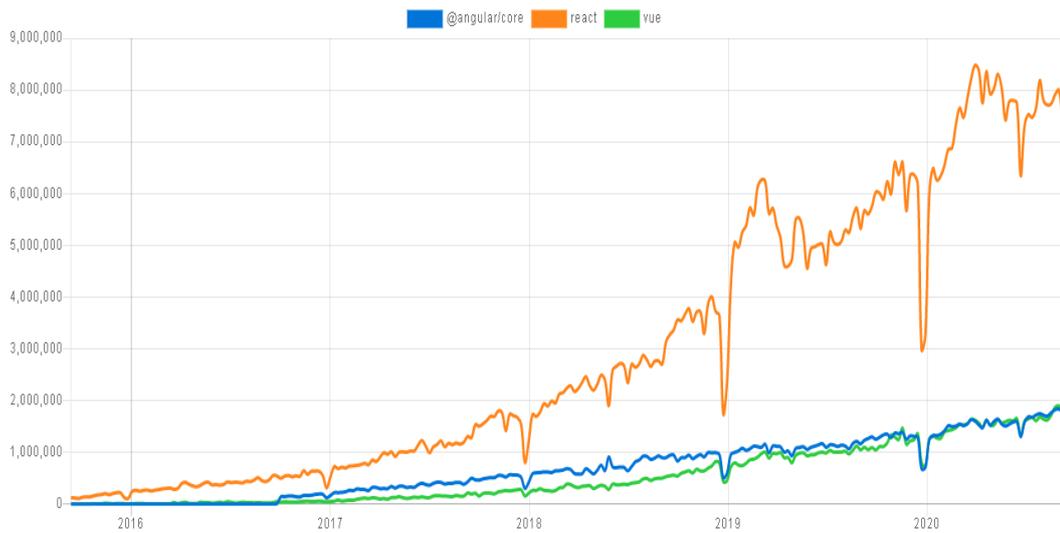


Figure 4.2: NPM Trends Results [46]

4.1.3 Google Trends

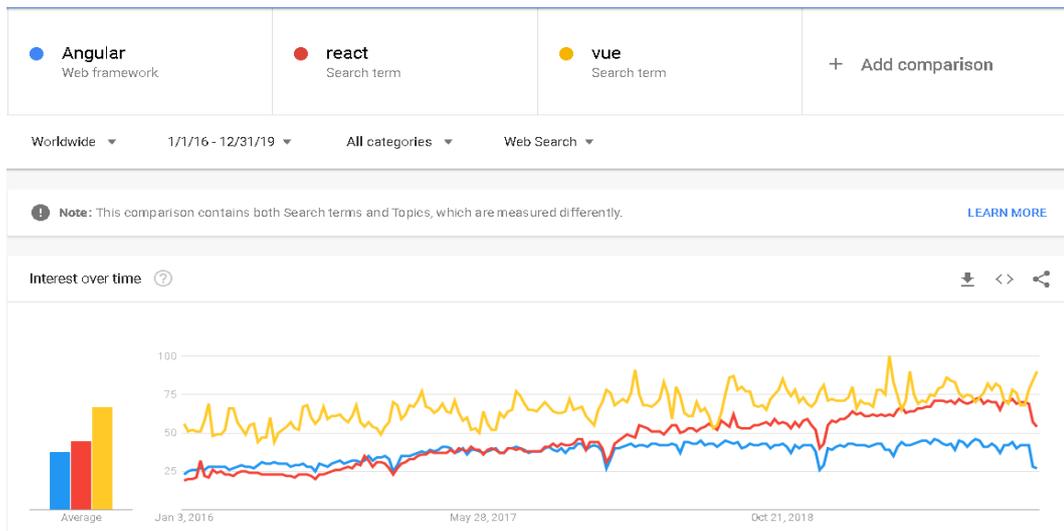


Figure 4.3: Google Trends Results [47]

Figure [4.3] shows the popularity of keywords worldwide on Google from 2016 to 2019. The popularity of 100 is peak popularity, and 50 is half popularity, and 0 means there was not enough data [47]. The keyword “vue” has the highest popularity score most of the time of around 90 at the end of 2019. The keyword “react” has a popularity of 54, and the keyword “angular” has a popularity of 27. These keywords

do not directly correspond to the JavaScript framework and are considered a regular term by the search engine.

4.1.4 Answering RQ-1

React.js, Vue.js, and Angular are already the most wanted web frameworks according to the Stack Overflow annual developer survey 2019, as shown in Figure 3.1. We studied the download statistics, Google search trends, and Stack Overflow trends of these frameworks. React.js turned out to be the most downloaded framework and most asked tag on Stack Overflow, followed by Angular and Vue.js; Whereas Vue.js is the most popular keyword on Google search engine, followed by React.js and Angular.

4.2 Learning Factor on Stack Overflow

- **RQ-2:** How is Stack Overflow helping developers learn JavaScript frameworks?

4.2.1 Categorizing the Data

Manual Classification

JS Framework	Manual Classification	
Angular	Learning	4
	Review	89
Reactjs	Learning	4
	Review	115
Vue.js	Learning	0
	Review	105

Table 4.1: Manual Classification Results - Top 500 Posts

Table 4.1 shows the result of manual classification, which is considered the ground truth for the automated classification of the top 500 posts. The result shows the number of posts classified in the question category of “Review” and “Learning” for each JavaScript framework. Angular and React.js have 4 posts classified into the “Learning” category. Angular has 89 posts, and React.js has 115 posts classified into the “Review” category. Vue.js has 105 posts classified into the “Review” category and has no posts classified into the “Learning” category. We skip the learning category for Vue.js for automated classification, as there were no posts classified into the “Learning” category. As this result is considered as the ground truth, the performance metrics for the same is assumed to be 100 % accurate. Precision, Recall, and Matthews Correlation Coefficient (MCC) is considered to be 1.

Automated Classification

JS Framework	Automated Classification	
Angular	Learning	1
	Review	32
Reactjs	Learning	1
	Review	41
Vue.js	Learning	0
	Review	32

Table 4.2: Automated Classification Results - Random 150 Posts

Table 4.2 shows the result of the automated classification of random 150 posts. This automated classification is carried out with the help of the regex approach. The result shows the number of posts classified in the question category of “Review” and “Learning” for each JavaScript framework. Angular and React.js have 1 post classified into the “Learning” category. Angular has 32 posts, and React.js has 41 posts classified into the “Review” category. Vue.js has 32 posts classified into the “Review” category.

4.2.2 Performance Metrics

JS Framework	Automated Classification	Precision	Recall	MCC
Angular	Learning	1	1	1
	Review	1	0.82	0.87
Reactjs	Learning	1	1	1
	Review	0.97	0.85	0.87
Vue.js	Learning	-	-	-
	Review	1	0.88	0.92

Table 4.3: Performance Metrics - Automated Classification

Table 4.3 shows the performance metrics of automated classification results using the regex approach to classify posts into question categories. Posts with the tag “Angular” is classified into the correct question category with an average precision, recall, and MCC of 1, 0.91, and 0.93, respectively. Posts with the tag “Reactjs” is classified into the correct question category with an average precision, recall, and MCC of 0.98, 0.92, and 0.93, respectively. Posts with the tag “Vue.js” is classified into the correct question category with precision, recall, and MCC of 1, 0.88, and 0.92, respectively. In this scenario, higher precision is preferred over recall because we aim to classify posts arguing that we do not classify a post instead of classifying posts in a wrong question category [12]. A higher precision states that the posts were correctly classified into question categories.

4.2.3 Answering RQ-2

Initially, we manually classified the Stack Overflow questions into question categories to find the ground truth for our automated classification. The ground truth is used as a base for automated classification using the regex approach. Considering the “Learning” and “Review” category as one category, just “Learning,” the automated classifier classified 33 posts for Angular, 42 posts for React.js, and 32 posts for Vue.js from an independent dataset of 150 questions with an average precision of 0.99. Angular and Vue.js have the same amount of learning content on Stack Overflow, while React.js has a slightly higher learning content on Stack Overflow. The presence of learning content on Stack Overflow indicates that apart from just asking and answering questions related to JavaScript frameworks, Stack Overflow also helps the community members to learn and get familiar with these JavaScript frameworks with the content being generated. Henceforth, it is extremely beneficial for the developers to learn and get familiar with JavaScript frameworks while visiting these Stack Overflow posts, which promotes learning.

4.3 University Students in Germany on Stack Overflow

- **RQ-3:** How are university students in Germany learning JavaScript frameworks on Stack Overflow?

4.3.1 Stack Overflow Annual Developer Survey Results

Student Status

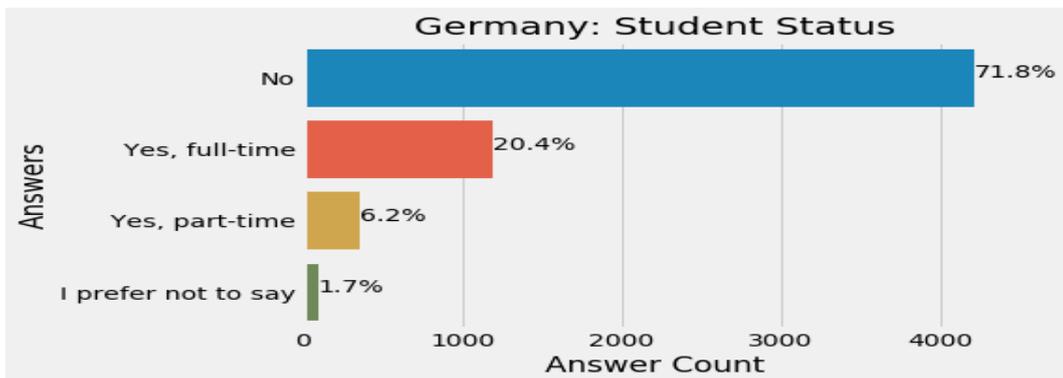


Figure 4.4: Student Status

Figure [4.4](#) shows the participant’s status concerning studying at a university in Germany or working professionally. 20.4 % percent of the participants are full-time students in Germany, and 6.2 % are part-time students in Germany. We consider

4 Results

20.4 % full-time and 6.2 % part-time student responses for further analysis to look at student's thought processes.

Student Age Group

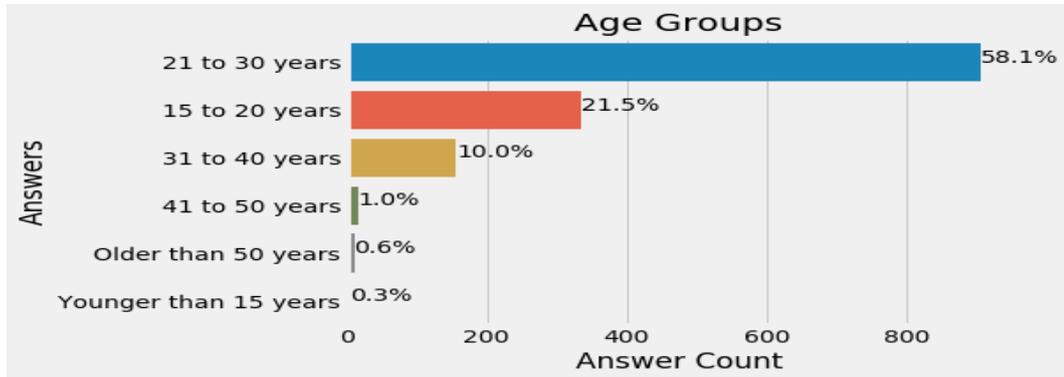


Figure 4.5: Student Age Group

Figure 4.5 shows the age group of students who participated in the survey. 58.6 % of the students are between the age of 21 to 30 years. Even students from the age group of 15-20 years constitute 21.5 % of the participants.

Student Education Level

Figure 4.6 shows the education level of students who participated in the survey. The majority of the students, about 36.8 % have only completed a secondary level of education. 29.7 % of students have a bachelor's degree, and 15.5 % of students have a master's degree. 10.8 % went to a university but did not get a degree yet. 2.6 % have an associate degree, 1.0 % have an other doctoral degree, 0.9 % have a professional degree, 0.4 % have a primary/elementary school education, and 0.2 % never completed any formal education.

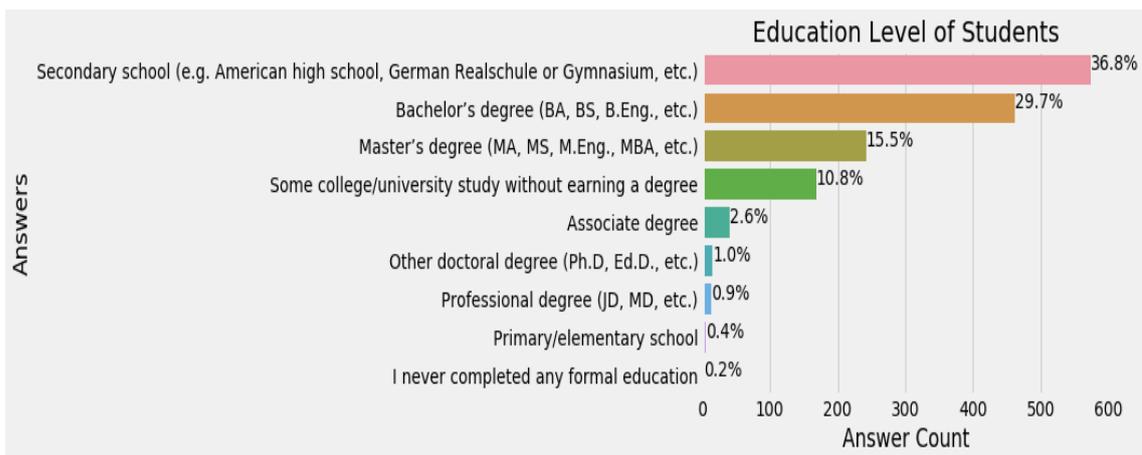


Figure 4.6: Student Education Level

Student Undergraduate Major

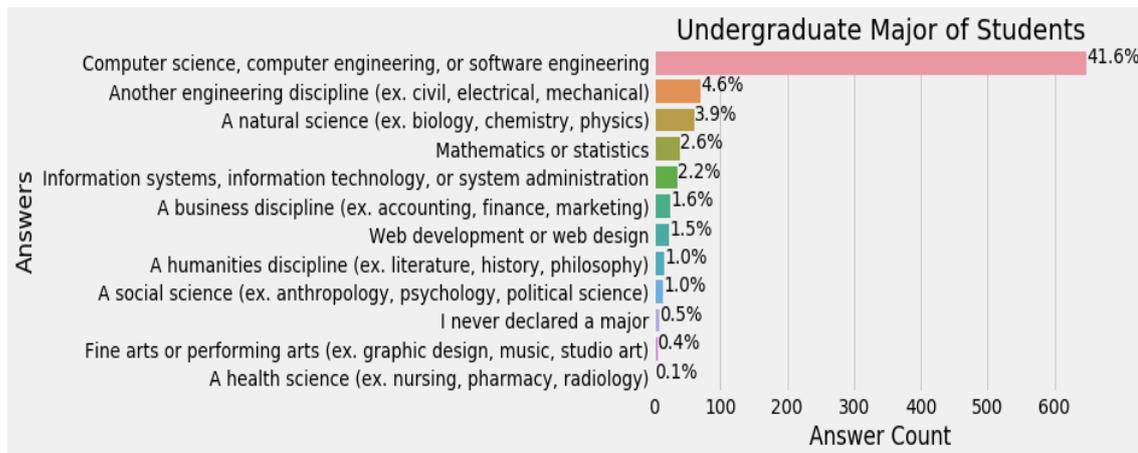


Figure 4.7: Student Undergraduate Major

Figure 4.7 shows the major specialization in the students who are doing or completed their undergraduate degrees. 41.6 % of students have chosen their specialization in computer science in their undergraduate degree.

Programming Languages used by Students

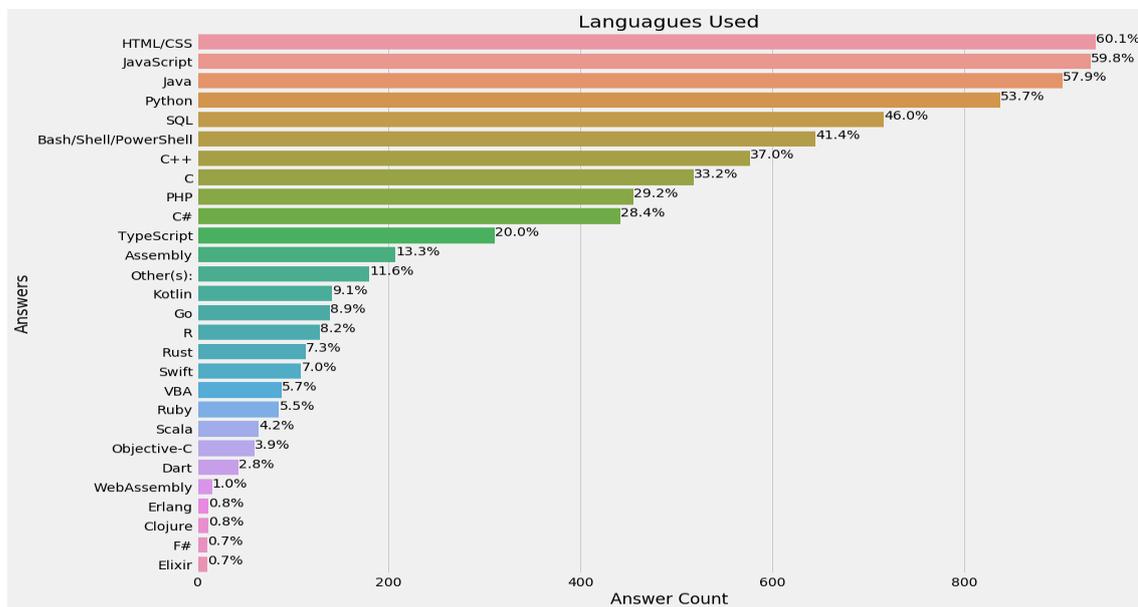


Figure 4.8: Programming Languages used by Students

4 Results

Figure 4.8 shows the programming languages used by the students from Germany who participated in the survey. The top programming languages used by students are HTML/CSS, JavaScript, Java, Python with 60.1 %, 59.8 %, 57.9 %, and 53.7 % respectively.

Programming Languages desired by Students

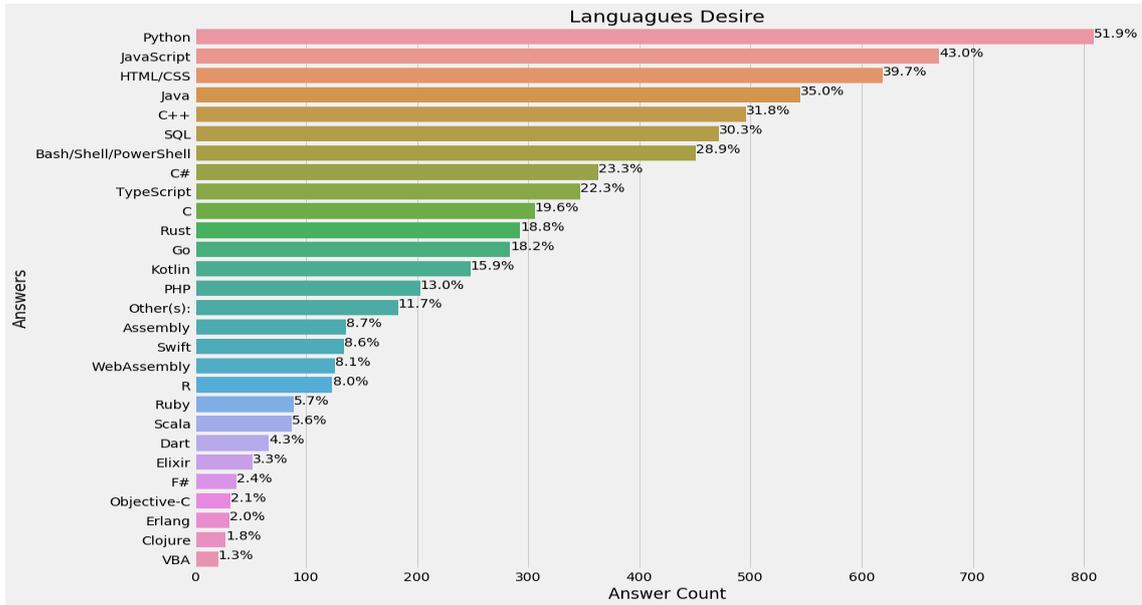


Figure 4.9: Programming Languages desired by Students

Figure 4.9 shows the programming languages desired by the students to work with in the future from Germany. The top programming languages desired by students are Python, JavaScript, HTML/CSS, Java with 51.9 %, 41 %, 39.7 % and 35 % respectively.

Web Frameworks used by Students

Figure 4.9 shows the web frameworks used by the students from Germany who participated in the survey. The top web frameworks used by students are jQuery, Angular/Angular.js, React.js, Spring, Express with 29.5 %, 18.2 %, 15.5 %, 12.8 %, and 12.1 % respectively. 4 out of these 5 frameworks are JavaScript frameworks.

4 Results

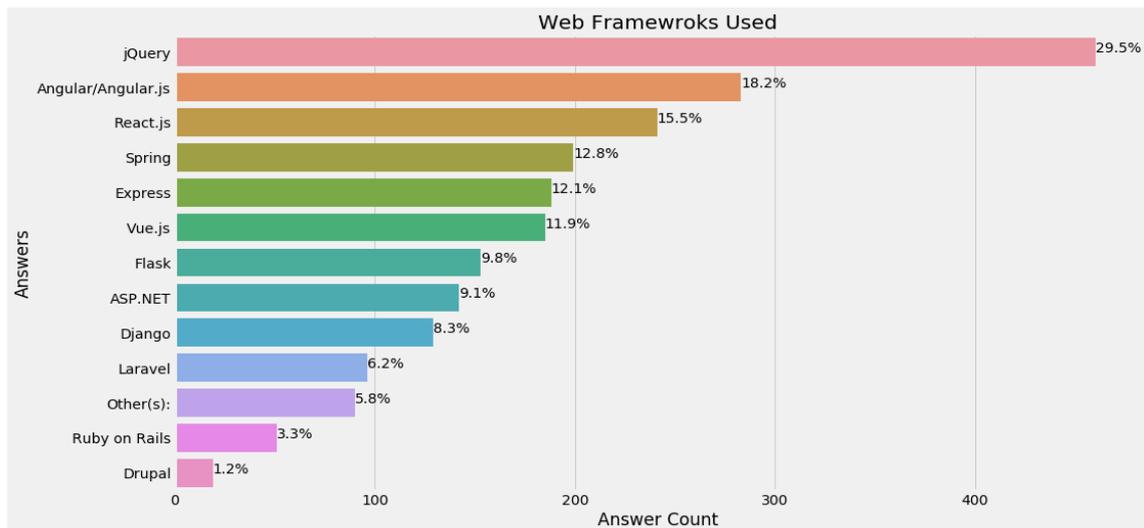


Figure 4.10: Web Frameworks used by Students

Web Frameworks desired by Students

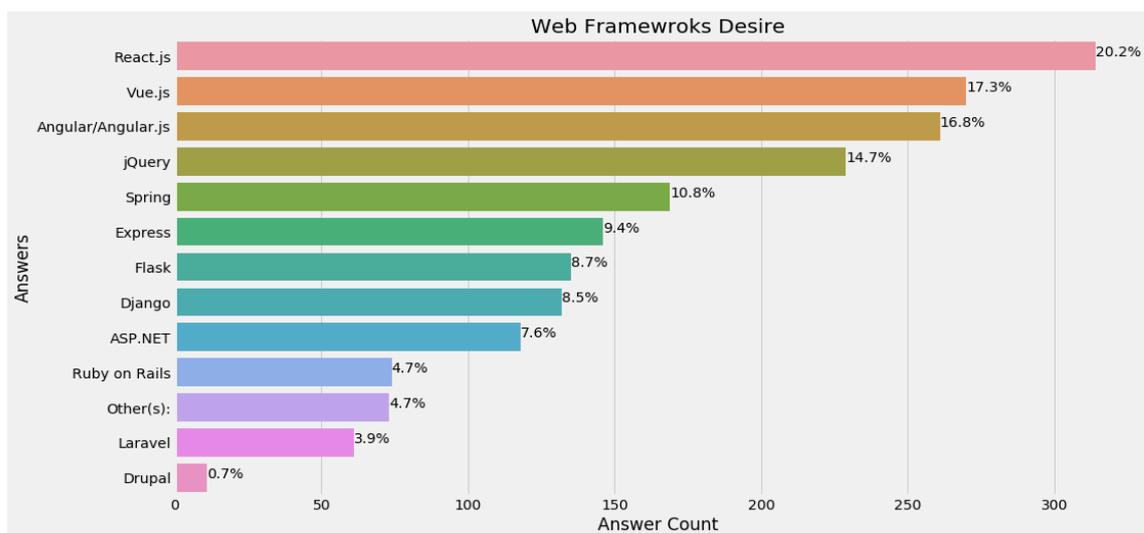


Figure 4.11: Web Frameworks desired by Students

Figure [4.11](#) shows the web frameworks desired by the students work with in the future from Germany who participated in the survey. The top web frameworks desired by students are React.js, Vue.js, Angular/Angular.js, jQuery, Spring with 20.2 %, 17.3 %, 16.8 %, 14.7 %, and 10.8 %, respectively. 4 out of these 5 frameworks are also JavaScript frameworks.

Why do students visit Stack Overflow?

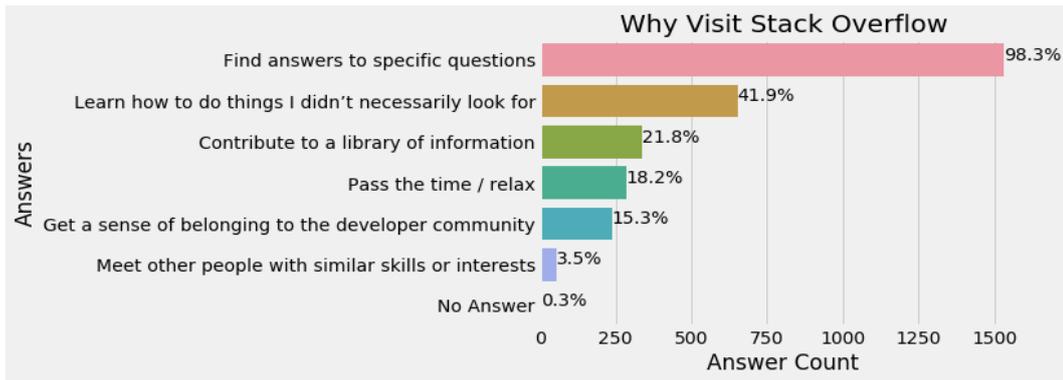


Figure 4.12: Why Students visited Stack Overflow

Figure 4.12 shows the list of reasons why students visit Stack Overflow. Students were allowed to check more than one answer. 98.3 % of students came to Stack Overflow to find answers to specific questions. Other reasons were not so much used to visit Stack Overflow.

Opinions on new content types

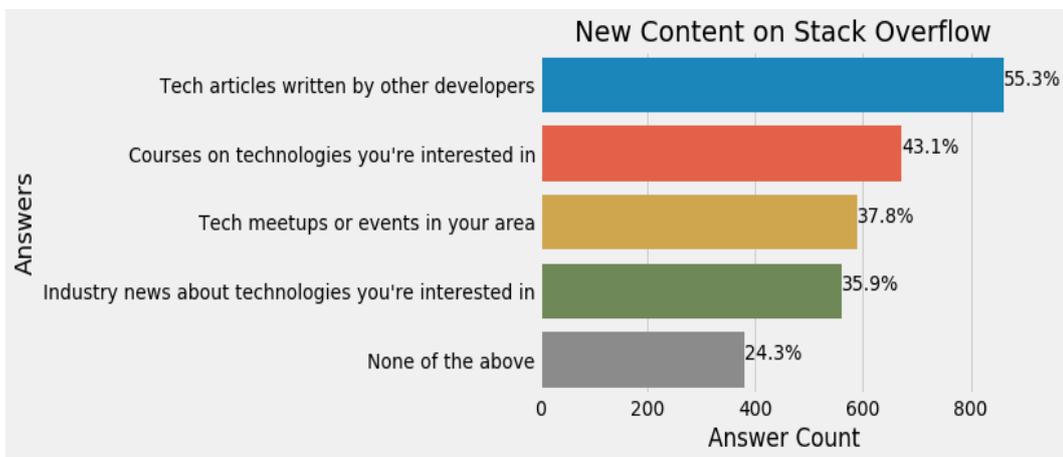


Figure 4.13: Opinions on new content types

Figure 4.13 shows the opinions of students they want to see on Stack Overflow. About 55.3 % would like to see technical articles written by other developers, and the least exciting topic is industry news about technologies. 24.3 % of students said they don't want to see anything new on Stack Overflow.

4.3.2 Student's Survey Result

Participant's Background

Are you a student at the Chemnitz University of Technology?

72 responses

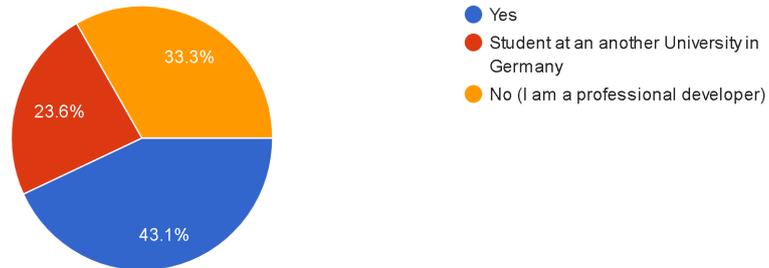


Figure 4.14: Participant's Background

Figure 4.14 shows the participant's status concerning studying at the Chemnitz University of Technology or a university in Germany or working professionally. 43.1 % of the participants are students at the Chemnitz University of Technology, and 23.6 % are students at another university in Germany.

Web Development Experience

Do you have any experience working with Web Development?

72 responses

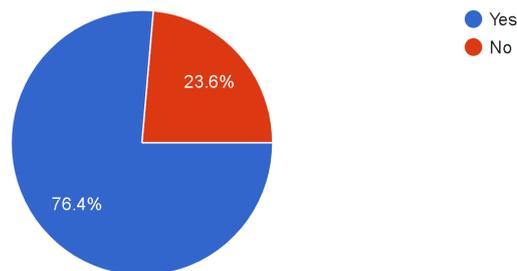


Figure 4.15: Web Development Experience

Figure 4.15 shows the web development experience of participants. 76.4 % of participants have experience in web development.

Interest/Experience in JavaScript Frameworks

Do you have any experience or interest in learning at least one of the Javascript Frameworks or libraries? (e.g. Angular, Reactjs, Vuejs, Nodejs etc)
72 responses

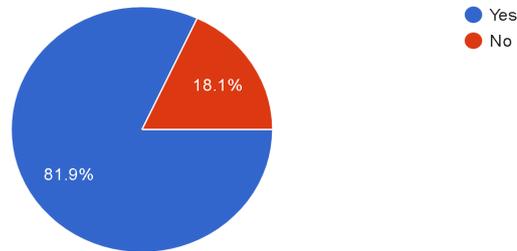


Figure 4.16: Interest/Experience in JavaScript frameworks

Figure 4.16 shows the experience with JavaScript frameworks of participants. 81.9 % of participants have experience and interest in JavaScript frameworks.

Stack Overflow Usage

Have you ever used a Q&A platform like Stack Overflow to solve your coding problems?
72 responses

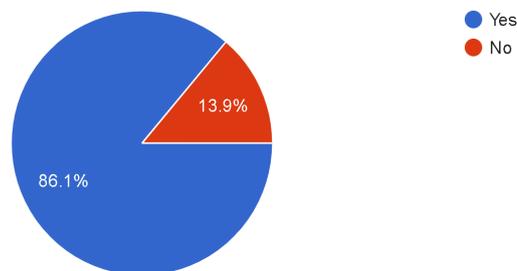


Figure 4.17: Stack Overflow usage

Figure 4.17 shows the usage of Stack Overflow for solving coding problems of participants. 86.1 % of participants have used Stack Overflow to solve coding problems.

Helpfulness on Stack Overflow

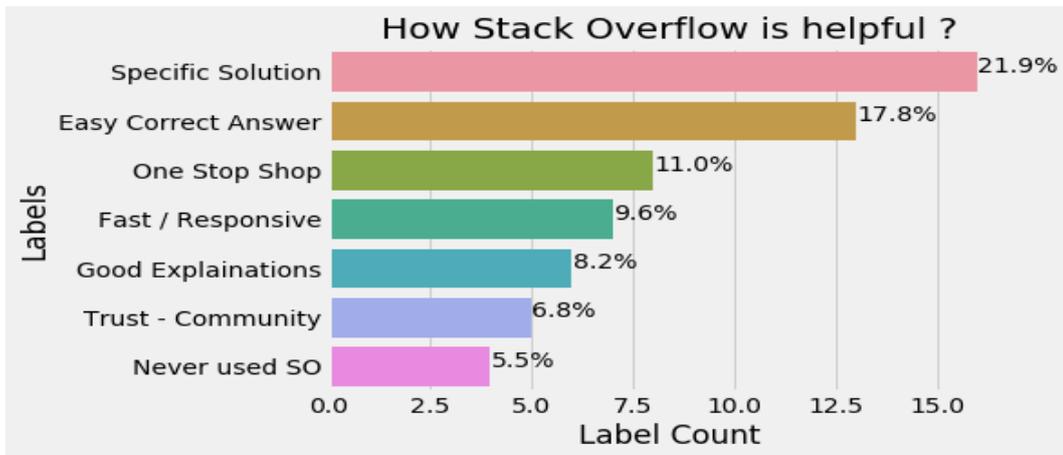


Figure 4.18: Helpfulness on Stack Overflow

Figure 4.18 shows labels of how Stack Overflow helped participants compared to a different source. The answers were the free thoughts of the participants. This result information is processed using an open card sort method and presented as labels. The majority of the participants, about 21.9 % said Stack Overflow provided a specific solution to them, while 6.8 % of participants said they trust the Stack Overflow community.

Learning with Stack Overflow

Do you think you can learn a programming language or any framework/library with the help of Stack Overflow?
72 responses

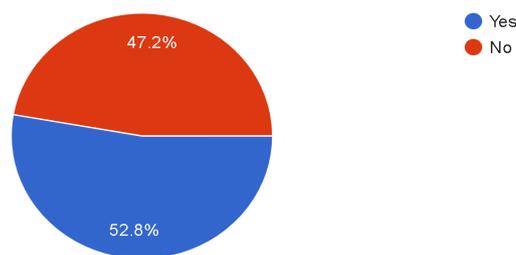


Figure 4.19: Learning with Stack Overflow

Figure 4.19 shows the percentage of participants whether they can learn a programming language or any framework with the help of Stack Overflow. 52.8 % of

4 Results

participants said they could learn with the help of Stack Overflow. On the other hand, 47.2 % of participants said they could not learn with Stack Overflow.

Rating the Learning Experience

On a scale of 1 - 5. How much will you rate your learning experience on Stack Overflow
72 responses

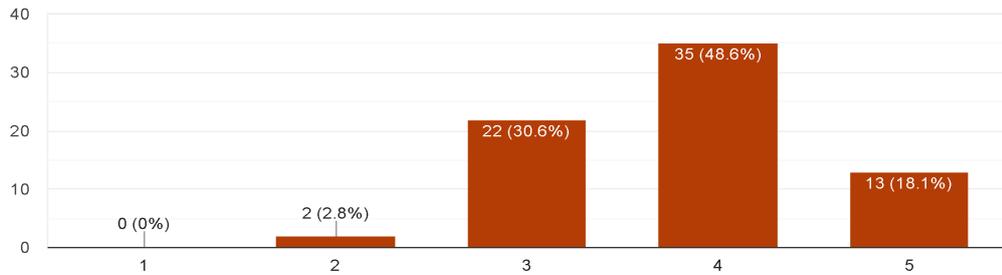


Figure 4.20: Rating the Learning Experience

Figure 4.20 shows participant's self-evaluation for their learning experience on Stack Overflow on a scale of 1 to 5 where 1 is Very Dissatisfied, 2 is Dissatisfied, 3 is Neutral, 4 is Satisfied, and 5 is Very Satisfied. 48.6 % of participants are satisfied, and 30.6 % of participants are neutral about their learning experience, while 2.8 % are dissatisfied.

Challenges faced on Stack Overflow

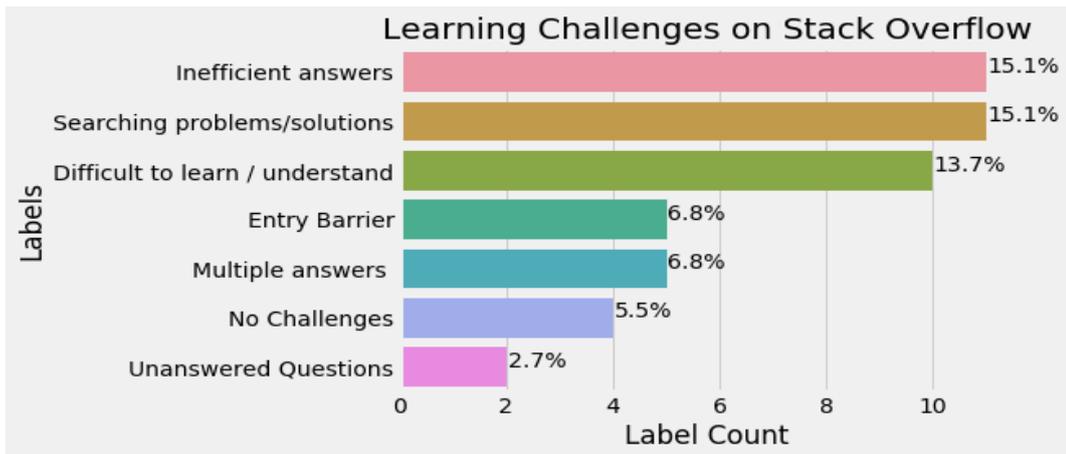


Figure 4.21: Challenges faced on Stack Overflow

Figure 4.21 shows the labels of what challenges participants face on Stack Overflow while learning a new language or a framework. The answers were the free thoughts of the participants. This result information is processed using an open card sort method and presented as labels. 15.1 % of participants said that inefficient answers and searching problems or solutions are the most faced challenges on Stack Overflow. 13.5 % said it is difficult to understand the answers, while 6.8 % of participants said there is an entry barrier and Stack Overflow has multiple answers. 2.7 % said unanswered questions are the challenges they faced.

Stack Overflow: Learning Tool?

Do you think Stack Overflow can be a valuable resource/tool for learning?
72 responses

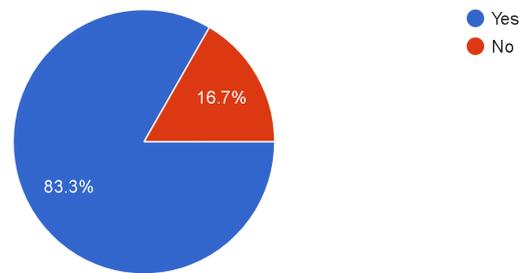


Figure 4.22: Stack Overflow: Learning Tool?

Figure 4.22 shows whether the participants think Stack Overflow can be a valuable tool or not. 83.3 % of participants think that Stack Overflow can be a valuable tool for learning.

Improvements on Stack Overflow

Figure 4.23 shows the improvements suggested by participants for Stack Overflow to become a Q&A based learning platform. The answers were the free thoughts of the participants. This result information is processed using an open card sort method and presented as labels. Most of the participants, about 17.8 % do not require any improvements on Stack Overflow. 15.1 % of participants would like to see the questions on Stack Overflow in a classified way. 11 % of participants would like to see technical articles and videos. 6.8 % of participants would like to remove inefficient answers and improve answering standards on Stack Overflow.

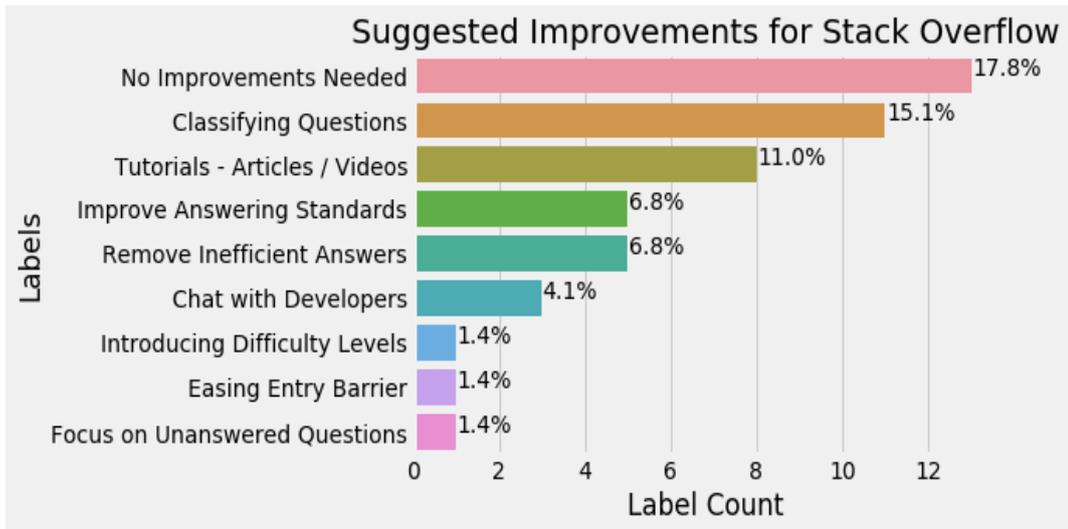


Figure 4.23: Improvements on Stack Overflow

Rank Learning Methodology

When learning a programming language or a framework/library, how will you rank your learning methodology?

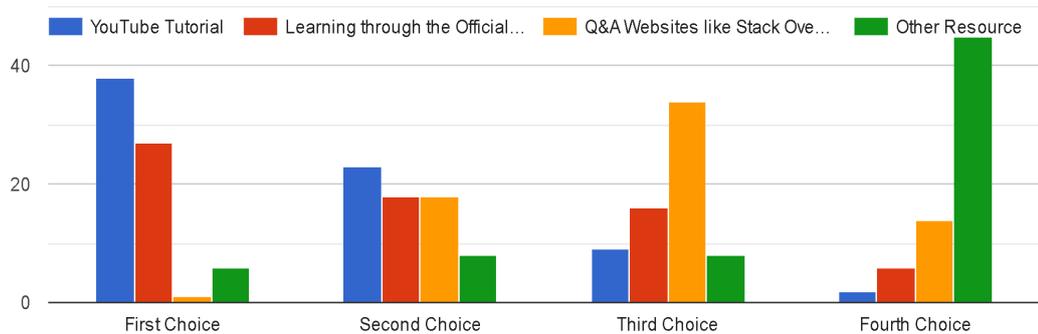


Figure 4.24: Rank Learning Methodology

Figure 4.24 shows the self-evaluation rankings of participants of their learning methodology for learning a programming language. YouTube tutorials and official documentation are the most preferred place to learn a new programming language or a framework.

4.3.3 Answering RQ-3

Students in Germany mostly desire to work with Python programming language and React.js web framework. Students mostly visit Stack Overflow to find answers to their specific questions. Only half of them visit to learn things they do not need. Students would like to see tech articles written on Stack Overflow to gain knowledge in various technologies. Challenges faced by students are inefficient answers on Stack Overflow. Students suggest the questions on Stack Overflow should be presented in classified form into different categories. A majority of the students from universities in Germany, especially students from the Chemnitz University of Technology, considers SO a valuable resource for learning JavaScript frameworks and are very satisfied with their learning experience on Stack Overflow. However, YouTube is the first choice for students to learn JavaScript frameworks, followed by official documentation and Stack Overflow. Learning JavaScript frameworks on Stack Overflow is beneficial for students, but it comes with some challenges, such as inefficient answers.

5 Discussion

In this section, we first summarize and discuss our results.

5.1 Interpretation of Results

- **RQ-1:** What are the most in-demand JavaScript frameworks in 2019?

Our results state that in 2019, React.js is the most downloaded JavaScript framework, followed by Angular and Vue.js with similar download statistics. Most of the questions on Stack Overflow are asked for React.js, followed by Angular and Vue.js by the end of 2019. Our results confirmed that React.js, as Brandon Satrom [17] and Eric Wohlgethan [18] reported, is the most downloaded JavaScript framework. According to our results, Vue.js has almost similar download statistics compared to Angular. However, questions asked on Stack Overflow for Vue.js are far less than the questions asked for Angular. Considering lower questions asked for Vue.js while having similar download statistics as Angular, these statistics support the studies reported by Brandon Satrom [17], Eric Wohlgethan [18], Elar Saks [19], Nicklas Ockelberg, and Niclas Olsson [20] that Vue.js has the smallest learning curve compared to Angular and React.js. Also, Vue.js is the most popular googled keyword compared to the other two frameworks. Our results for download statistics and asked questions, indicates that in 2019, React.js is the most in-demand JavaScript framework, followed by Angular and Vue.js. However, the continuous evolution of various JS frameworks can replace the most in-demand JS framework in 2019 with some other framework in the future based on its ability to draw the developer community's attention with a new set of features better than the existing framework.

- **RQ-2:** How is Stack Overflow helping developers learn JavaScript frameworks?

Our results support the method of classifying Stack Overflow posts into question categories by Stefanie Beyer, Christian Macho, Martin Pinzger, and Massimiliano Di Penta [12]. We combined the question category “Review” and “Learning” as “Learning,” because the questions and answers classified in the review category have more of a learning content in them. On a sample of 150 random questions for each framework, the automated classifier based on the ground truth from manual classification classified 33 posts for Angular, 42 posts for React.js, and 32 posts for Vue.js for the “Learning” category with an average precision of 0.99. Based on our

results, we can state that Angular and Vue.js have the same amount of learning content on Stack Overflow, while React.js has a slightly higher amount of learning content on Stack Overflow. Almost a quarter of our independent dataset promotes learning content indicating that Stack Overflow helps developers learn JavaScript frameworks mentioned in the thesis. The presence of learning content on Stack Overflow suggests that apart from just asking and answering questions related to programming, SO also helps the community members to learn and get familiar with programming, enhancing their problem-solving skills. Developers can benefit from spending time on Stack Overflow to learn some new concepts. Still, it is subjective for each developer on which answer the developer is concentrating more on, an efficient solution with a high score or an inefficient solution with a low score. While Angular and Vue.js have the same amount of learning content on Stack Overflow based on our dataset, it somehow contradicts the claims by Brandon Satrom [17], Eric Wohlgethan [18], Elar Saks [19], Nicklas Ockelberg, and Niclas Olsson [20] that Vue.js has the smallest learning curve and Angular has the steepest learning curve. The presence of learning content on Stack Overflow also indicates that these JavaScript framework’s official documentation might be expressed in a complicated way that is not easy for developers to understand. The JavaScript frameworks can simplify their official documentation because answers on Stack Overflow will not always be efficient enough, which can negatively impact developers.

- **RQ-3:** How are university students in Germany learning JavaScript frameworks on Stack Overflow?

Around 1558 students from universities in Germany participated in the Stack Overflow annual developer survey 2019 [5]. The participants are majorly from the computer science background, mostly between the age group of 15-30. The top web frameworks participants desired to work with are React.js, Vue.js, and Angular. The majority of the participants visit Stack Overflow to find answers to their specific questions and would like to see articles and courses by other developers in the future. We surveyed some open questions with students from universities in Germany, majorly from the Chemnitz University of Technology. Majorly the participants were from a web development background and had an interest or experience with JavaScript frameworks. The participants found Stack Overflow helpful for specific solutions to questions and the availability of easy and correct answers. This finding supports the Stack Overflow annual developer survey result on why participants visit Stack Overflow. The most common challenges faced by participants on Stack Overflow are inefficient answers and the considerable effort to search for a problem or a solution. Lastly, they faced difficulty in understanding the answers. The majority of the participants think Stack Overflow can be a valuable tool for learning and suggested some improvements for Stack Overflow. The participants indicated that the questions on Stack Overflow should be classified into question categories so that it is easy to find a problem or solution. They also suggested they would like to see

articles and videos of different technologies on Stack Overflow. Some other suggestion includes removing inefficient answers and improving the answering standards. Learning on Stack Overflow can be useful for students. Still, it can positively or negatively impact them based on the efficiency of the answers, because the most common challenges faced by the participants were inefficient answers available on Stack Overflow. Stack Overflow helps students learn to program, but it does not guarantee logical thinking enhancement due to inefficient answers. The learning experience is also subjective to every individual user as well.

6 Threats to Validity

Threats to construct validity concerns the use of Stack Overflow to learn JavaScript frameworks. The construct is measured by classifying Stack Overflow posts into question categories, especially the “Learning” and “Review” category. The measure of this construct is entirely accurate, with an average classifier precision of 0.99.

Threats to internal validity concern the selection of Stack Overflow posts for manual classification. 500 posts were selected, which had an accepted answer, and their question score and answer count were greater than their average. These posts had 92 % confidence and 8 % margin of error. Manual classification can be biased because a single researcher carried it out. There can be a difference of opinions between researchers about classifying a post into a specific question category, resulting in the different ground truth used for automated classification. The results of automated classification are drawn from a sample size of 150 random posts. Results may vary depending on the sample size, but it is still useful to get a glimpse inside the diverse data found on Stack Overflow. Another threat to internal validity concerns is interpreting the participant’s answers to the survey’s open questions. There is no ground truth for analyzing this kind of data. We still interpreted the data using the open card sort technique. We classify similar data under the same category in the open card sort technique and represent it quantitatively. The controversial answers were carefully examined to minimize any potential classification error.

Threats to external validity concern the generalizability of our results for the interpretation of data on Stack Overflow. The posts retrieved from Stack Overflow are related to tags “angular,” “reactjs,” and “vue.js.” Learning content on Stack Overflow can differ for different JavaScript frameworks on Stack Overflow. The results are useful to study the data of only these JSF on Stack Overflow. Another threat to external validity is the recruited participant’s opinions in the survey. For the open questions, participants expressed their views regarding Stack Overflow. The survey was carried out online using Google forms. The kind of communication and the environment can differ from face-to-face interviews, but an online survey helps understand the developer community’s thought process.

7 Conclusion

This thesis investigated the most in-demand JavaScript frameworks in 2019 and how Stack Overflow helps developers learn JavaScript frameworks. We also analyzed the various experience of students from Germany, especially from the Chemnitz University of Technology, while learning JavaScript frameworks with Stack Overflow.

As a first step, we identified the most in-demand JavaScript framework by analyzing the most downloaded framework on NPM and the most searched keywords among these frameworks on Google search engine. We also studied the frequency of questions asked on Stack Overflow using Stack Overflow trends, which can be the deciding factor for a particular framework’s difficulty to learn. Based on our quantitative analysis of the JavaScript frameworks, it can be concluded that React.js is the most in-demand JavaScript framework in 2019, followed by Angular and Vue.js. The results indicated that Vue.js has the smallest learning curve with the least number of questions asked on Stack Overflow with almost similar download statistics to that of other frameworks.

As a second step, we analyzed the data of Angular, React.js, and Vue.js on Stack Overflow by categorizing them into different question categories, especially the “Learning” and “Review” category. Initially, manual classification is carried out on Stack Overflow posts with a high score and accepted answer metrics. Later, an automated classification is carried out on independent datasets using regex and keywords from the manual classification. The classifier has an average precision of 0.99. The results confirm the presence of learning content on Stack Overflow. The results indicated that apart from just asking and answering questions related to JavaScript frameworks, Stack Overflow also helps the community members to learn and get familiar with these JavaScript frameworks.

As a third step, we analyzed the responses of full-time and part-time students from the Stack Overflow annual developer survey 2019. We also surveyed students from Germany, especially from the Chemnitz University of Technology, and analyzed their responses. Students in Germany mostly desire to work with Python programming language and React.js web framework. Students mostly visit Stack Overflow to find answers to their specific questions. Students would like to see tech articles written on Stack Overflow to gain knowledge in various technologies. A majority of the students from universities in Germany, especially students from the Chemnitz University of Technology, considers Stack Overflow a valuable resource for learning JavaScript frameworks and are very satisfied with their learning experience on Stack Overflow. However, YouTube is the first choice for students to

learn JavaScript frameworks, followed by official documentation and Stack Overflow. Learning JavaScript frameworks on Stack Overflow is beneficial for students, but it comes with some challenges, such as inefficient answers.

The overall research contributes towards showcasing the rising trends within the growing JavaScript framework ecosystem. Further, it highlights the presence of learning content on Stack Overflow for JavaScript frameworks such as Angular, React.js, and Vue.js. It also highlights the thought processes of a small amount of student population in Germany, which can be considered the forthcoming generations entering the workforce. Stack Overflow helps to develop problem-solving skills as it benefits developers and students to get familiar with programming. The learning experience on Stack Overflow does not always benefit the user due to inefficient answers on Stack Overflow. Still, Stack Overflow can be regarded as a viable platform for developers and students to develop new creative ideas solving real-world problems.

For future work, the rising trends in JavaScript frameworks can be calculated based on their real-time applications. The tech-stack of real-time applications can be found using StackShare [48] and Wappalyzer [49]. JavaScript frameworks should consider simplifying their official documentation as the thesis found learning content on Stack Overflow, which ideally should not be the case. Classification of Stack Overflow questions can help recommender systems based on Stack Overflow, such as Seahawk [50] and Prompter [51]. For researchers, the research about learning content can be extended to a large amount of data spread across various other programming languages and web frameworks. A new study on the impact of learning from Stack Overflow based on the efficiency of answers can be initiated. For developers, a tool or a plugin can be a viable option to present classified questions of Stack Overflow.

8 Acknowledgments

I have received a significant amount of help and assistance in the writing of this thesis. I would like to thank Prof. Dr. Janet Siegmund, and Shadi Saleh, my supervisors, whose experience in the formulation of research questions and methods was invaluable. I was pushed by your constructive suggestions to sharpen my analysis and to carry my work to a higher level. Prof. Siegmund continuously encouraged and was always willing and enthusiastic to assist in any way she could work throughout the thesis. I would like to thank the participants who took part in the study and enabled this research to be possible.

I would also like to thank my parents for their sound guidance and caring ears. For me, you are still there. Finally, without the help of my friends, Karan Shukla and Karan Jagad, and my wife, Arushi, I could not have completed this thesis. To rest my mind outside of my study, they offered stimulating conversations as well as happy distractions.

Bibliography

- [1] “Usage statistics of JavaScript as client-side programming language on websites.” (Visited On: 07/2020). [Online]. Available: <https://w3techs.com/technologies/details/cp-javascript/>
- [2] J. Spolsky, “Stackoverflow.com.” April 2008, (Visited On: 07/2020). [Online]. Available: <https://www.joelonsoftware.com/2008/04/16/stackoverflowcom/>
- [3] J. Atwood, “Help Name Our Website.” April 2008, (Visited On: 07/2020). [Online]. Available: <https://blog.codinghorror.com/help-name-our-website/>
- [4] “Stack Exchange.” (Visited On: 07/2020). [Online]. Available: <https://stackexchange.com/sites?view=list>
- [5] “Stack Overflow Developer Survey 2019.” (Visited On: 07/2020). [Online]. Available: <https://insights.stackoverflow.com/survey/2019>
- [6] S. Beyer, C. Macho, M. Pinzger, and M. D. Penta, “Automatically classifying posts into question categories on stack overflow.” *Proceedings of the 26th Conference on Program Comprehension - ICPC '18.*, pp. 211–221, 2018, (Visited On: 08/2020). [Online]. Available: <https://dl.acm.org/doi/10.1145/3196321.3196333>
- [7] C. Rosen and E. Shihab, “What are mobile developers asking about? a large scale study using stack overflow,” *Empirical Software Engineering*, vol. 21, no. 3, pp. 1192–1223, 2016.
- [8] M. Allamanis and C. Sutton, “Why, when, and what: analyzing stack overflow questions by topic, type, and code,” *2013 10th Working Conference on Mining Software Repositories (MSR)*, pp. 53–56, 2013, (Visited On: 09/2020). [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/6624004>
- [9] C. Treude, O. Barzilay, and M.-A. Storey, “How do programmers ask and answer questions on the web?(nier track),” *Proceedings of the 33rd international conference on software engineering*, pp. 804–807, 2011, (Visited On: 09/2020). [Online]. Available: <https://dl.acm.org/doi/abs/10.1145/1985793.1985907>
- [10] S. Beyer and M. Pinzger, “A manual categorization of android app development issues on stack overflow,” *2014 IEEE International Conference on Software Maintenance and Evolution*, pp. 531–535, 2014, (Visited On: 09/2020). [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/6976132>

BIBLIOGRAPHY

- [11] S. Beyer, C. Macho, M. Di Penta, and M. Pinzger, “Analyzing the relationships between android api classes and their references on stack overflow,” *Technical Report*, 2017, (Visited On: 09/2020). [Online]. Available: <https://serg.aau.at/pub/StefanieBeyer/Publications/AAU-SERG-techreport.pdf>
- [12] S. Beyer, C. Macho, M. Pinzger, and M. D. Penta, “What kind of questions do developers ask on stack overflow? a comparison of automated approaches to classify posts into question categories.” *Empirical Software Engineering*, pp. 2258–2301, 2020, (Visited On: 08/2020). [Online]. Available: <https://doi.org/10.1007/S10664-019-09758-X>
- [13] I. Allen, “The brutal lifecycle of javascript frameworks.” 2019, (Visited On: 09/2020). [Online]. Available: <https://stackoverflow.blog/2018/01/11/brutal-lifecycle-javascript-frameworks/>
- [14] S. Hjelm, “Evaluation of javascript frameworks: Why should you use them?” 2015, (Visited On: 09/2020). [Online]. Available: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A867126&dswid=-309>
- [15] “Understanding client-side javascript frameworks.” (Visited On: 09/2020). [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks
- [16] A. Pano, D. Graziotin, and P. Abrahamsson, “Factors and actors leading to the adoption of a javascript framework,” *Empirical Software Engineering*, vol. 23, no. 6, pp. 3503–3534, 2018, (Visited On: 09/2020). [Online]. Available: <https://doi.org/10.1007/s10664-018-9613-x>
- [17] B. Sotom, “Choosing the right javascript framework for your next web application,” *Prog./Kendo UI*, p. 34, 2018, (Visited On: 09/2020). [Online]. Available: <https://www.telerik.com/docs/default-source/whitepapers/telerik-com/choose-the-right-javascript-framework-for-your-next-web-application-whitepaper.pdf>
- [18] E. Wohlgethan, “Supporting web development decisions by comparing three major javascript frameworks: Angular, react and vue. js,” 2018, (Visited On: 09/2020). [Online]. Available: https://edoc.sub.uni-hamburg.de/haw/volltexte/2018/4350/pdf/BA_Wohlgethan_2176410.pdf
- [19] E. Saks, “Javascript frameworks: Angular vs react vs vue.” 2019, (Visited On: 09/2020). [Online]. Available: <https://www.theseus.fi/bitstream/handle/10024/261970/Thesis-Elar-Saks.pdf?sequence=2>
- [20] N. Olsson and N. Ockelberg, “Performance, modularity and usability, a comparison of javascript frameworks,” 2020, (Visited On: 09/2020). [Online]. Available: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1424374&dswid=-9520>

BIBLIOGRAPHY

- [21] D. Robinson, “*Introducing Stack Overflow Trends.*” May 2017, (Visited On: 07/2020). [Online]. Available: <https://stackoverflow.blog/2017/05/09/introducing-stack-overflow-trends/>
- [22] “*Stack Overflow Trends.*” (Visited On: 08/2020). [Online]. Available: <https://insights.stackoverflow.com/trends?tags=r%2Cstatistics>
- [23] “*About npm.*” (Visited On: 07/2020). [Online]. Available: <https://docs.npmjs.com/about-npm/>
- [24] “*NPM Trends: Compare NPM package downloads.*” (Visited On: 08/2020). [Online]. Available: <https://www.npmtrends.com/>
- [25] “*Google Trends.*” (Visited On: 08/2020). [Online]. Available: <https://trends.google.com/trends/?geo=US>
- [26] “*Search Engine Market Share Worldwide.*” (Visited On: 07/2020). [Online]. Available: <https://gs.statcounter.com/search-engine-market-share>
- [27] “*Where Developers Learn, Share & Build Careers.*” (Visited On: 08/2020). [Online]. Available: <https://stackoverflow.com/?products>
- [28] “*Tour - Stack Overflow.*” (Visited On: 07/2020). [Online]. Available: <https://stackoverflow.com/tour>
- [29] J. Atwood, “*Stack Overflow Creative Commons Data Dump.*” June 2009, (Visited On: 07/2020). [Online]. Available: <https://stackoverflow.blog/2009/06/04/stack-overflow-creative-commons-data-dump/>
- [30] —, “*Now Licensed Under Creative Commons.*” May 2008, (Visited On: 07/2020). [Online]. Available: <https://stackoverflow.blog/2008/05/05/now-licensed-under-creative-commons/>
- [31] D. Fullerton, “*Stack Exchange Creative Commons data now hosted by the Internet Archive.*” January 2014, (Visited On: 07/2020). [Online]. Available: <https://stackoverflow.blog/2014/01/23/stack-exchange-cc-data-now-hosted-by-the-internet-archive/>
- [32] J. Atwood, “*Introducing Stack Exchange Data Explorer.*” June 2010, (Visited On: 07/2020). [Online]. Available: <https://stackoverflow.blog/2010/06/13/introducing-stack-exchange-data-explorer/>
- [33] “*Help - Stack Exchange Data Explorer.*” (Visited On: 07/2020). [Online]. Available: <https://data.stackexchange.com/help>
- [34] K. Pike, “*You Can Now Play with Stack Overflow Data on Google’s BigQuery.*” December 2016, (Visited On: 07/2020). [Online]. Available: <https://stackoverflow.blog/2016/12/15/you-can-now-play-with-stack-overflow-data-on-googles-bigquery/>

BIBLIOGRAPHY

- [35] F. Hoffa, “Google bigquery public datasets now include stack overflow q&a.” 2016, (Visited On: 09/2020). [Online]. Available: <https://stackoverflow.blog/2018/01/11/brutal-lifecycle-javascript-frameworks/>
- [36] “*Query Stack Overflow.*” (Visited On: 08/2020). [Online]. Available: <https://data.stackexchange.com/stackoverflow/query/new>
- [37] D. Chicco, “Ten quick tips for machine learning in computational biology,” *BioData Mining*, vol. 10, no. 1, 2017, (Visited On: 08/2020). [Online]. Available: <https://biodatamining.biomedcentral.com/articles/10.1186/s13040-017-0155-3>
- [38] D. M. W. Powers, “Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation,” *International Journal of Machine Learning Technology*, vol. 2, no. 1, pp. 37–63, 2011, (Visited On: 08/2020). [Online]. Available: <http://hdl.handle.net/2328/27165>
- [39] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*. Springer-Verlag Berlin Heidelberg, 2008. [Online]. Available: <https://doi.org/10.1007/978-3-540-76917-0>
- [40] “pandas - python data analysis library,” (Visited On: 09/2020). [Online]. Available: <https://pandas.pydata.org/>
- [41] “Numpy,” (Visited On: 09/2020). [Online]. Available: <https://numpy.org/>
- [42] “Matplotlib: Python plotting,” (Visited On: 09/2020). [Online]. Available: <https://matplotlib.org/>
- [43] “seaborn: statistical data visualization,” (Visited On: 09/2020). [Online]. Available: <https://seaborn.pydata.org/>
- [44] J. Nielsen, “Usability testing of www designs,” 1995, (Visited On: 08/2020). [Online]. Available: <https://www.nngroup.com/articles/usability-testing-1995-sun-microsystems-website/>
- [45] “*Stack Overflow Trends Result.*” (Visited On: 08/2020). [Online]. Available: <https://insights.stackoverflow.com/trends?tags=angular%2Creactjs%2Cvue.js>
- [46] “*NPM Trends Result.*” (Visited On: 08/2020). [Online]. Available: <https://www.npmtrends.com/@angular/core-vs-react-vs-vue>
- [47] “*Google Trends Result.*” (Visited On: 08/2020). [Online]. Available: <https://trends.google.com/trends/explore?date=2016-01-01%202019-12-31&q=%2Fg%2F11c6w0ddw9,react,vue>
- [48] “Software and technology stacks used by top companies,” (Visited On: 09/2020). [Online]. Available: <https://stackshare.io/>

BIBLIOGRAPHY

- [49] “Technology driven lead generation,” (Visited On: 09/2020). [Online]. Available: <https://www.wappalyzer.com/>
- [50] L. Ponzanelli, A. Bacchelli, and M. Lanza, “Seahawk: Stack overflow in the ide,” *2013 35th International Conference on Software Engineering (ICSE)*, pp. 1295–1298, 2013, (Visited On: 09/2020). [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/6606701/>
- [51] L. Ponzanelli, G. Bavota, M. Di Penta, R. Oliveto, and M. Lanza, “Prompter: A self-confident recommender system,” *2014 IEEE International Conference on Software Maintenance and Evolution*, pp. 577–580, 2014, (Visited On: 09/2020). [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/6976143/>

Name: Vorname: geb. am: Matr.-Nr.:	Bitte beachten: 1. Bitte binden Sie dieses Blatt am Ende Ihrer Arbeit ein.
---	--

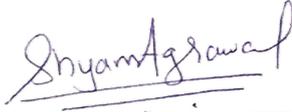
Selbstständigkeitserklärung*

Ich erkläre gegenüber der Technischen Universität Chemnitz, dass ich die vorliegende selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum:

Unterschrift: 

* Statement of Authorship

I hereby certify to the Technische Universität Chemnitz that this thesis is all my own work and uses no external material other than that acknowledged in the text.

This work contains no plagiarism and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources.

This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published.