PROJECT REPORT

Autonomous Tagging Of Stack Overflow Questions

Name	Team Id
Paramasivam N	NM2023TMID06421
Karthik M	
Muthukumar S	
Aswin A	
Murugaraj G	

INDEX

INTRODUCTION

1.1 Project Overview

With the advent of online education, question and answer forums are becoming an increasingly popular resource for information. Examples include Stack Exchange, Quora, and forums for Massive Open Online Courses (MOOCs) like Coursera and OpenEdX. While the quantity of information available on these forums is steadily increasing, there is currently no efficient and automatic way of grouping and classifying the information so that it can be displayed to users in an intuitive way. It would be useful to automatically infer and tag the topic of a question posted on a forum. A system that automatically infers the topic of a question can improve user-experience on online forums by 1) grouping questions about common topics together for users to browse and 2) showing users' posts related to a question they are inputting, since their question may already have been answered on the forum. In order to allow the grouping of common posts, some forums such as Quora require users to manually enter tags associated with their questions. However, manually tagging a post is a burden for users which degrades the overall user-experience. We envision a platform that can infer the tags of posts automatically. To this end, we propose a multi-label classification system that automatically assigns tags for questions posted on a forum. We implement and test our classifier on a dataset of Stack Overflow questions. The remainder of this paper is organized as follows. Section 2 outlines previous work on text classification, Section 3 describes our methodology and provides system architecture details, including feature extraction and classifier tuning. We present results in Section 4 and discuss key insights in Section 5, concluding in Section 6 with opportunities for future work.

1.2 PURPOSE

autonomous tagging of Stack Overflow questions is to automatically assign relevant tags to new questions posted on the platform. Stack Overflow is a popular online community for programmers and developers seeking assistance and sharing knowledge.Tagging plays a crucial role in organizing and categorizing the vast amount of questions and answers on the platform. It helps users filter and search for questions related to specific technologies, programming languages, frameworks, or concepts. By assigning appropriate tags to questions, it becomes easier for users to find relevant information and experts in the corresponding subject areas.Autonomous tagging aims to streamline the process of assigning tags by leveraging machine learning and natural language processing techniques. By analyzing the text of a question, including the title, body, and any accompanying code snippets, an autonomous tagging system can identify key terms and topics and match them to relevant tags from a predefined set. This helps improve the accuracy and consistency of tagging across the platform.

IDEATION & PROPOSED SOLUTION

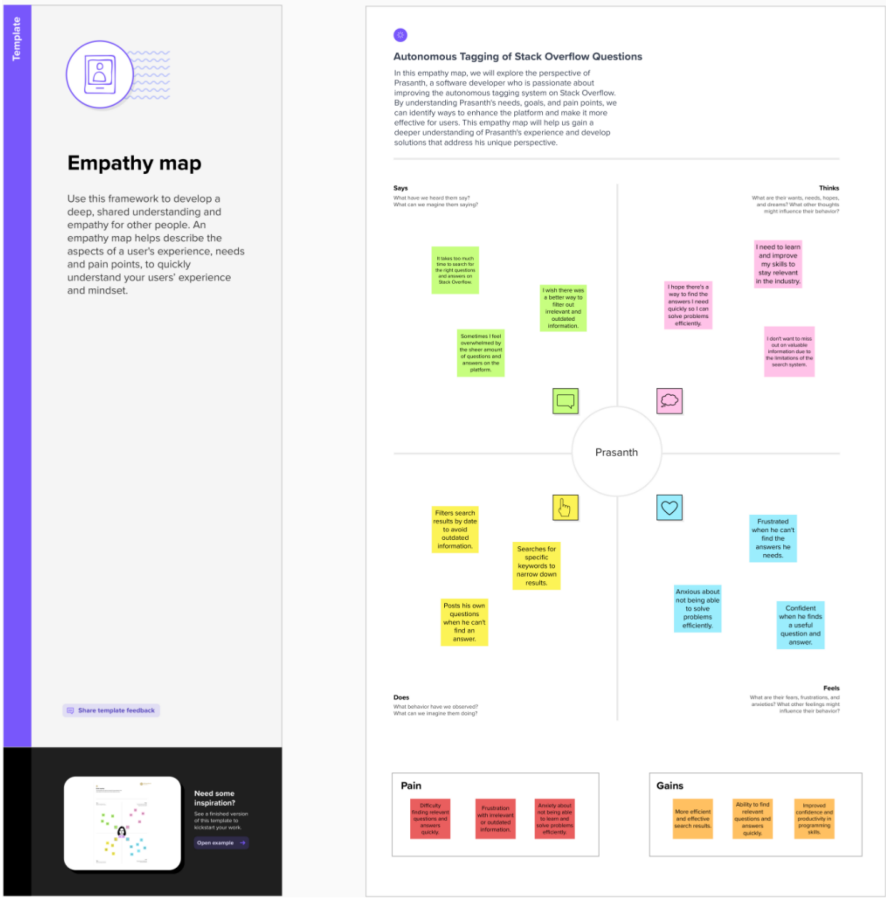
2.1 Problem Statement Definition

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Software Developer	learn a new programming language	it takes a lot of time to filter	of overwhelming number of questions and answers on Stack Overflow	frustrated

PS-2	not an expert in programming terminology	solve a specific coding problem	I cannot find relevant search results	of incomplete or inaccurate user-generated tags on Stack Overflow	stuck and demotivated
PS-3	relies heavily on Stack Overflow	find answers to my coding questions	I often struggle to find relevant information	**of inaccurate or incomplete tags**	Wasting time

2.2 Empathy Map Canvas

In this empathy map, we will explore the perspective of Prasanth, a software developer who is passionate about improving the autonomous tagging system on Stack Overflow. By understanding Prasanth's needs, goals, and pain points, we can identify ways to enhance the platform and make it more effective for users. This empathy map will help us gain a deeper understanding of Prasanth's experience and develop solutions that address his unique perspective.



Empathy map

Use this framework to develop a deep, shared understanding and empathy for other people. An empathy map helps describe the aspects of a user's experience, needs and pain points, to quickly understand your users' experience and mindset.

Need some inspiration?

See a finished version of this template to kickstart your work.

Autonomous Tagging of Stack Overflow Questions

In this empathy map, we will explore the perspective of Prasanth, a software developer who is passionate about improving the autonomous tagging system on Stack Overflow. By understanding Prasanth's needs, goals, and pain points, we can identify ways to enhance the platform and make it more effective for users. This empathy map will help us gain a deeper understanding of Prasanth's experience and develop solutions that address his unique perspective.

Says

What have we heard them say?
What can we magine them saying?

It takes too much time to search for the right questions and answers on Stack Overflow.

I wish there was a better way to filter out irrelevant and outdated information.

Sometimes I feel overwhelmed by the sheer amount of questions and answers on the platform.

Thinks

What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

I need to learn and improve my skills to stay relevant in the industry.

I hope there's a way to find the answers I need quickly so I can solve problems efficiently.

I don't want to miss out on valuable information due to the limitations of the search system.

Prasanth

Filters search results by date to avoid outdated information.

Searches for specific keywords to narrow down results.

Posts his own questions when he can't find an answer.

Frustrated when he can't find the answers he needs.

Anxious about not being able to solve problems efficiently.

Confident when he finds a useful question and answer.

Does

What behavior have we observed?
What can we imagine them doing?

Feels

What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

Pain

Difficulty finding relevant questions and answers quickly.

Frustration with irrelevant or outdated information.

Anxiety about not being able to learn and solve problems efficiently.

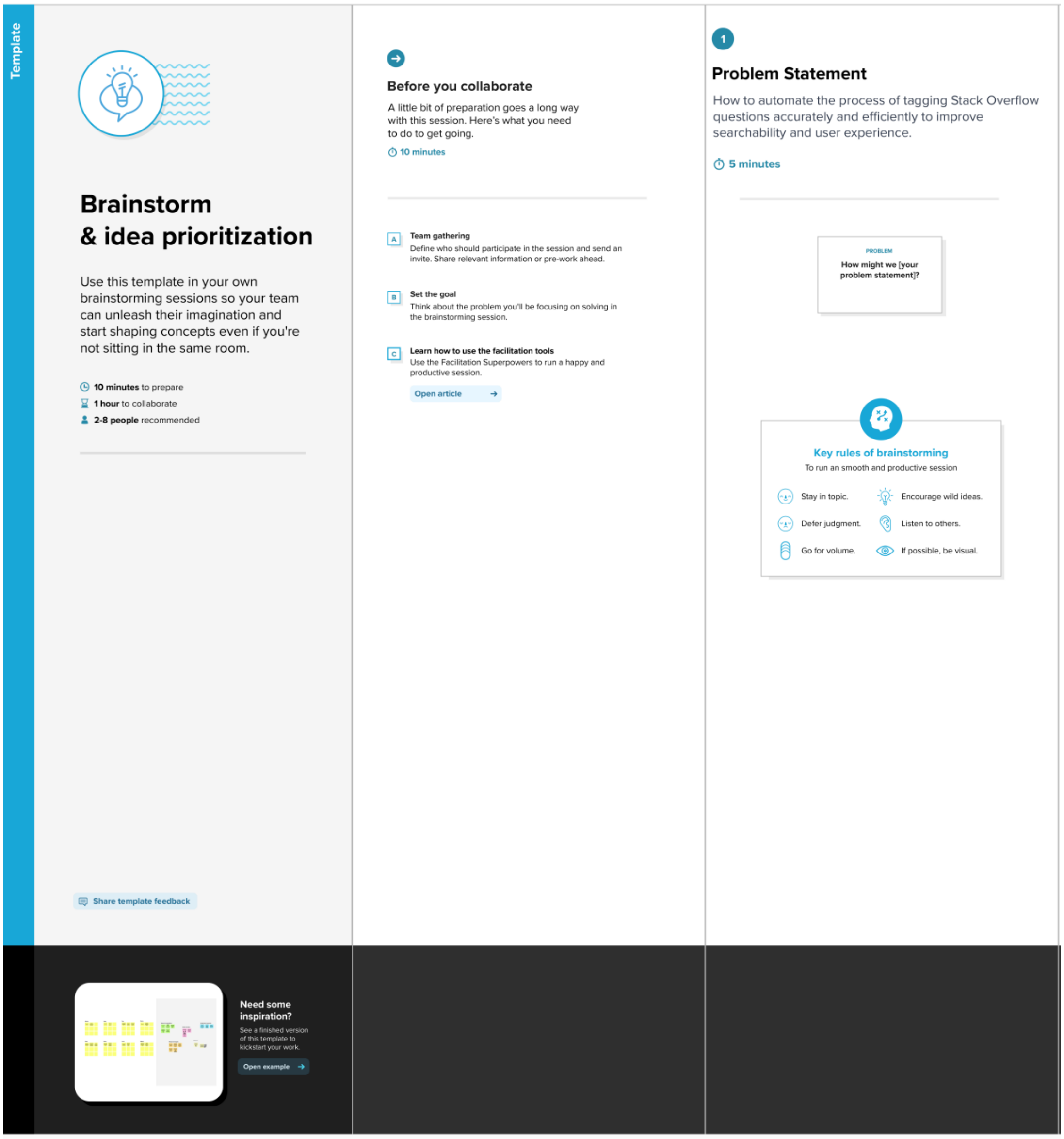
Gains

More efficient and effective search results.

Ability to find relevant questions and answers quickly.

Improved confidence and productivity in programming skills.

2.3 Ideation &Brainstorming



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Need some inspiration?

See a finished version of this template to kickstart your work.

Open example →

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Problem Statement

How to automate the process of tagging Stack Overflow questions accurately and efficiently to improve searchability and user experience.

5 minutes

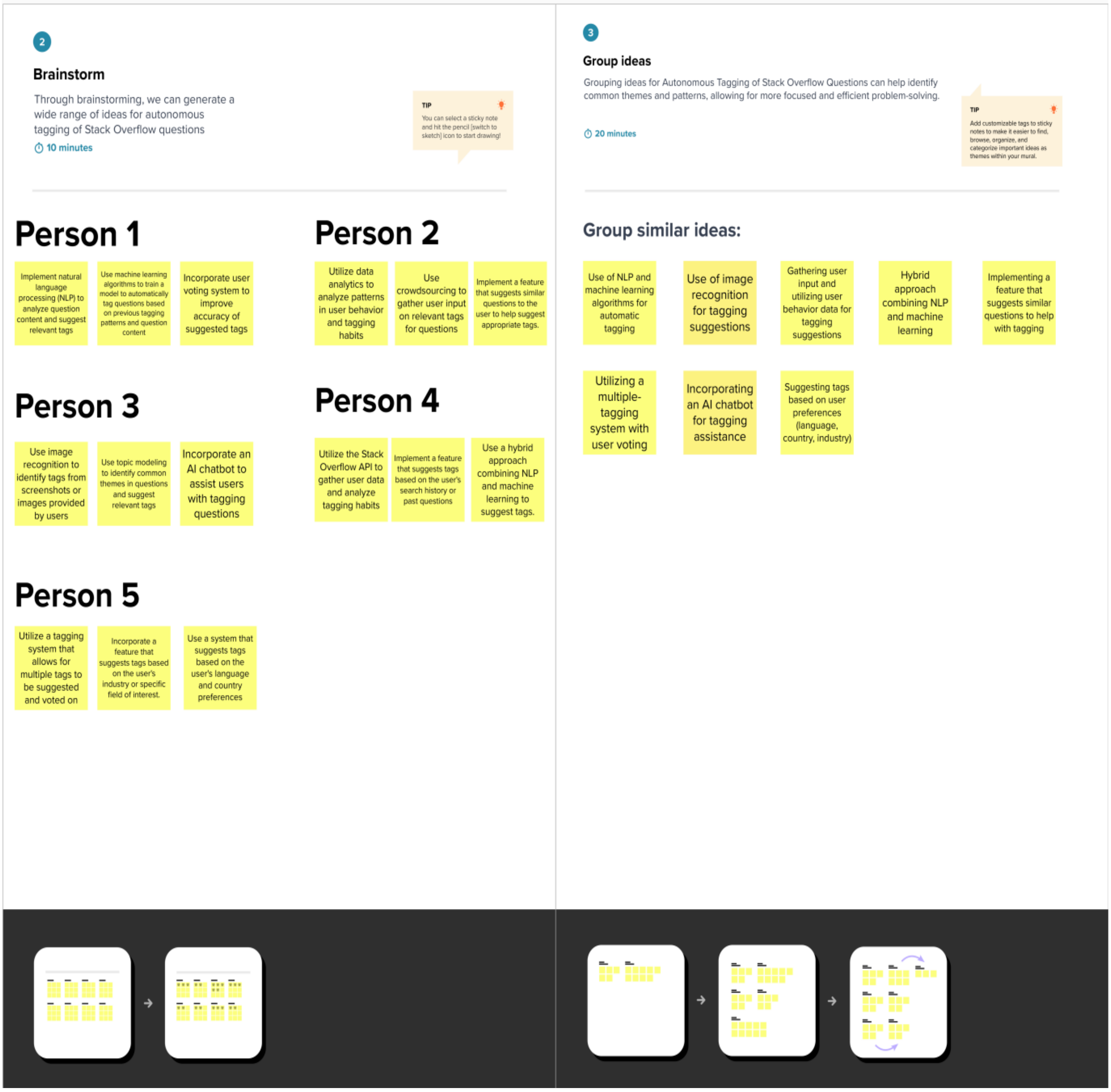
PROBLEM

How might we [your problem statement]?

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.



2

Brainstorm

Through brainstorming, we can generate a wide range of ideas for autonomous tagging of Stack Overflow questions

⏱ 10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Person 1

- Implement natural language processing (NLP) to analyze question content and suggest relevant tags
- Use machine learning algorithms to train a model to automatically tag questions based on previous tagging patterns and question content
- Incorporate user voting system to improve accuracy of suggested tags

Person 2

- Utilize data analytics to analyze patterns in user behavior and tagging habits
- Use crowdsourcing to gather user input on relevant tags for questions
- Implement a feature that suggests similar questions to the user to help suggest appropriate tags.

Person 3

- Use image recognition to identify tags from screenshots or images provided by users
- Use topic modeling to identify common themes in questions and suggest relevant tags
- Incorporate an AI chatbot to assist users with tagging questions

Person 4

- Utilize the Stack Overflow API to gather user data and analyze tagging habits
- Implement a feature that suggests tags based on the user's search history or past questions
- Use a hybrid approach combining NLP and machine learning to suggest tags.

Person 5

- Utilize a tagging system that allows for multiple tags to be suggested and voted on
- Incorporate a feature that suggests tags based on the user's industry or specific field of interest.
- Use a system that suggests tags based on the user's language and country preferences

3

Group ideas

Grouping ideas for Autonomous Tagging of Stack Overflow Questions can help identify common themes and patterns, allowing for more focused and efficient problem-solving.

⏱ 20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Group similar ideas:

- Use of NLP and machine learning algorithms for automatic tagging
- Use of image recognition for tagging suggestions
- Gathering user input and utilizing user behavior data for tagging suggestions
- Hybrid approach combining NLP and machine learning
- Implementing a feature that suggests similar questions to help with tagging
- Utilizing a multiple-tagging system with user voting
- Incorporating an AI chatbot for tagging assistance
- Suggesting tags based on user preferences (language, country, industry)

4

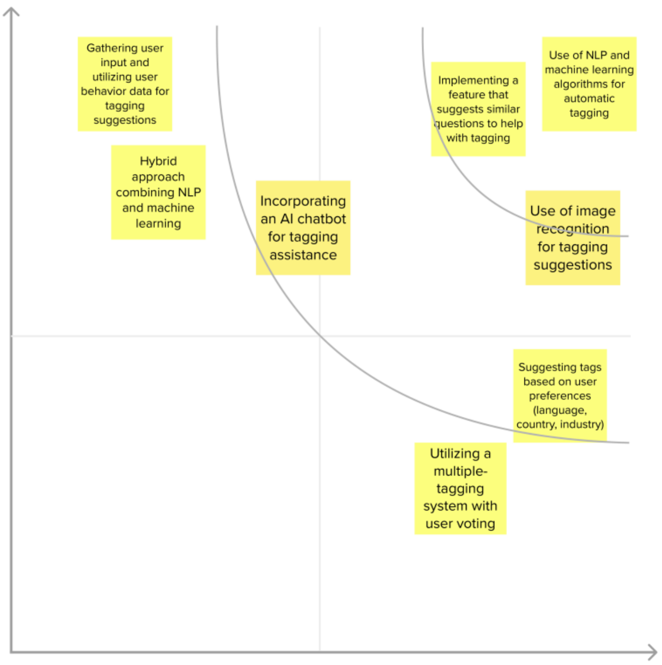
Prioritize

Prioritizing ideas involves assessing their importance and feasibility in order to determine which ones should be pursued first.

20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.



Gathering user input and utilizing user behavior data for tagging suggestions

Hybrid approach combining NLP and machine learning

Incorporating an AI chatbot for tagging assistance

Implementing a feature that suggests similar questions to help with tagging

Use of NLP and machine learning algorithms for automatic tagging

Use of image recognition for tagging suggestions

Suggesting tags based on user preferences (language, country, industry)

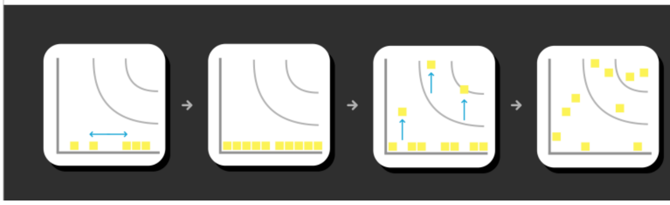
Utilizing a multiple-tagging system with user voting

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)



After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

2.4 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The vast number of questions and answers on Stack Overflow can make it difficult for individuals to filter through and find relevant content when finding answers to coding questions, resulting in reduced efficiency and frustration. Autonomous tagging of Stack Overflow questions can help address this issue by automatically categorizing questions based on their content, making it easier for users to find the information they need quickly and efficiently.
2.	Idea / Solution description	Autonomous tagging of Stack Overflow questions can be achieved through the use of machine learning algorithms that analyze the content of the questions and assign relevant tags automatically. This can be done by training the machine learning model on a dataset of labeled questions and tags. The model can then be used to predict tags for new questions. This approach can significantly improve the efficiency of finding relevant content on Stack Overflow, as users can easily search for questions by tags or browse questions with relevant tags. Additionally, users can contribute to the tagging process by suggesting tags, which can further improve the accuracy of the tagging system over time.
3.	Novelty / Uniqueness	The approach of autonomous tagging of Stack Overflow questions using machine learning algorithms is unique in its ability to categorize a large amount of data with accuracy and speed. This system can analyze the content of each question and assign appropriate tags automatically, which saves time and effort for users. The uniqueness of this approach also lies in its ability to continuously improve over time as users suggest more tags, resulting in an increasingly accurate tagging system. Overall, autonomous tagging offers a unique and innovative solution for improving the user experience on Stack Overflow.
4.	Social Impact / Customer Satisfaction	By making it easier for individuals to find the information they need quickly and efficiently, autonomous tagging can help democratize access to technical knowledge. This can benefit a wide range of people, including those from low-income backgrounds or those living in areas where access to formal education is limited. Additionally, by enabling more people to learn new programming languages, this approach can contribute to the development of a more technically skilled workforce, which can drive innovation and economic growth.
5.	Business Model (Revenue Model)	The revenue model for autonomous tagging of Stack Overflow questions could involve offering a subscription-based service to businesses or individuals who require fast and efficient access to relevant programming knowledge, or providing a

		paid API access to the autonomous tagging system for third-party developers and organizations.
6.	Scalability of the Solution	The solution of autonomous tagging of Stack Overflow questions through machine learning algorithms has high scalability potential, as it can process a large amount of data and handle a growing number of users and questions, making it a reliable and effective solution for improving the user experience on the platform.

Requirement Analysis

3.1 Function Requirement

Following are the functional requirements of the proposed solution for Autonomous Tagging Of Stack Overflow Questions

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Automatic Tagging	The system should be able to analyze the content of a Stack Overflow question and suggest tags based on the topic of the question.
FR-2	Machine Learning	The system should use machine learning algorithms to continually learn from user feedback and adjust the tagging process to improve accuracy over time.
FR-3	Review and Edit Tags	Moderators should be able to view and edit the tags suggested by the system before they are displayed publicly.
FR-4	Multi-Language Support	The system should be able to analyze questions written in multiple languages and suggest relevant tags in the same language as the question.
FR-5	Alternative Text for Images	The system should provide alternative text descriptions for images used in the tagging process, so that users with visual impairments or other disabilities can understand the content of the images. The descriptions should accurately reflect the content of the images.

3.2 Non-functional Requirements:

Following are the non-functional requirements of the proposed solution for Autonomous Tagging Of Stack Overflow Questions

FR No.	Non-Functional Requirement	Description

NFR-1	**Usability**	The system should be user-friendly and easy to use, with a clear and intuitive interface.
NFR-2	**Security**	The system should use secure protocols to protect user data and prevent unauthorized access.
NFR-3	**Reliability**	The system should be available and functional at all times, with minimal downtime and disruptions.
NFR-4	**Performance**	The system should be able to handle a large number of requests and users simultaneously, without slowing down or crashing.
NFR-5	**Availability**	The system should be available 24/7, with minimal scheduled downtime for maintenance and upgrades.
NFR-6	**Scalability**	The system should be able to scale up or down as needed to handle changing levels of traffic and usage.

Project Design

4.1 Data Flow Diagram

Data Flow Diagrams:

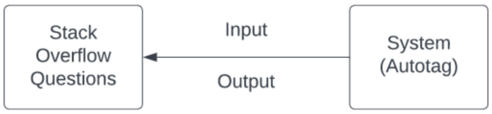
A Data Flow Diagram (DFD) is a graphical representation of the flow of data through a system. In the case of the Autonomous Tagging of Stack Overflow Questions system, a DFD would depict the flow of questions from Stack Overflow to the autonomous tagging system, where natural language processing (NLP) algorithms are used to analyze the text and identify relevant topics, and then the tagged questions are returned back to Stack Overflow for users to access.

The DFD for this system would include three main components: the Stack Overflow Questions database, the Autonomous Tagging System, and the Tagged Questions database. The DFD would show the flow of data between these components, as well as any processes or transformations that take place along the way.

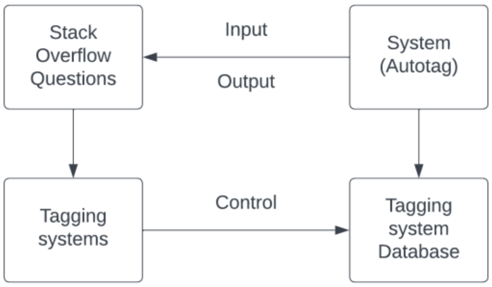
At the highest level, a Level 0 DFD would show the flow of questions from Stack Overflow to the Autonomous Tagging System, and then the flow of tagged questions back to Stack Overflow. At a more detailed Level 1 DFD, the processes involved in obtaining the questions, tagging them, and returning them would be broken down into separate components. Finally, a Level 2 DFD might show additional processes involved in the NLP algorithm used by the Autonomous Tagging System to identify relevant topics.

Data Flow Diagram (DFD) for Autonomous Tagging of Stack Overflow Questions:

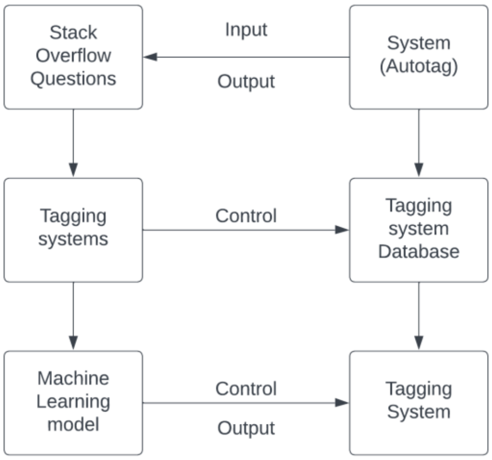
Level 0 DFD :



Level 1 DFD :

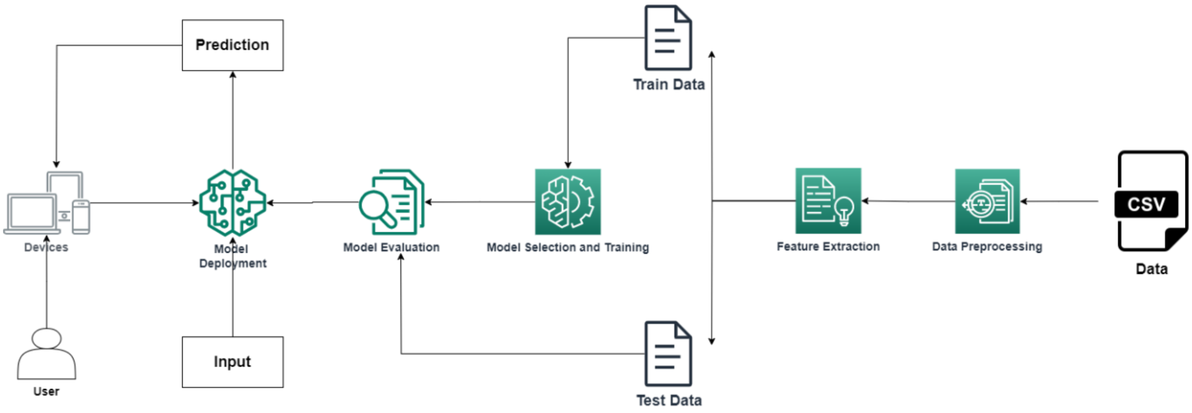


Level 2 DFD :



4.2 Solution &Technical Architecture

Architecture - Autonomous Stack Overflow Questions



Solution Architecture Description

Autonomous tagging of Stack Overflow questions can be achieved through a combination of natural language processing (NLP) and machine learning (ML) techniques. Here is a possible solution architecture for this problem:

1. Data Collection and Preprocessing: The first step is to collect the Stack Overflow questions data and preprocess it to remove any irrelevant information such as URLs, HTML tags, and special characters. This can be done using Python libraries like BeautifulSoup, pandas, and NLTK.

2. Feature Extraction: The next step is to extract the relevant features from the preprocessed data. This involves converting the text data into a numerical format that can be used by machine learning algorithms. Common feature extraction techniques include Bag of Words, TF-IDF, and Word Embeddings.

3. Machine Learning Model: Once the features are extracted, a machine learning model needs to be trained on the data to predict the appropriate tags for each question. A multi-label classification model can be used here, as each question can have multiple tags. Popular machine learning algorithms for multi-label classification include Random Forest, Naive Bayes, and Neural Networks.

4. Tagging and Evaluation: Once the model is trained, it can be used to tag new Stack Overflow questions with appropriate tags. The accuracy of the model can be evaluated using metrics like F1 score, precision, and recall. The model can be retrained and refined based on the evaluation results.

5. Deployment: Finally, the model can be deployed as a web service, allowing users to enter a new question and receive a list of relevant tags. This can be done using Python libraries like Flask or Django.

Overall, this solution architecture can provide an effective way to autonomously tag Stack Overflow questions using machine learning and natural language processing techniques**.**

4.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
Stack Overflow User	Automatic Tagging	USN-1	As a Stack Overflow user, I want my question to be automatically tagged based on its content	When a user submits a question, the system should analyze the question content and suggest tags that accurately represent the topic of the question. The user should be able to edit or remove any suggested tags as needed before submitting the question.	High	Paramasivam
Developer	Machine Learning	USN-2	As a developer, I want the system to use machine learning algorithms to improve the accuracy of the tagging process over time	The system should use a machine learning model that continually learns from user feedback and adjusts the tagging process to improve accuracy over time. The model should be transparent and easily auditable by developers and moderators.	High	Karthick
Moderator	Review and edit tags	USN-3	As a moderator, I want to be able to review and edit the tags suggested by the system	The system should allow moderators to view and edit the tags suggested by the system before they are displayed publicly. Moderators should be able to approve or reject suggested tags, as well as add or remove tags as needed.	High	Muthukumar
Non-English speaker	Multi-Language support	USN-4	As a non-English speaker, I want the system to support multiple languages	The system should be able to analyze questions written in multiple languages and suggest relevant tags in the same language as the question. The system should also support the ability for users to search for questions and tags in different languages.	Medium	Aswin
User with a Disability	Alternative Text for Images	USN-5	As a user with a disability, I want the system to provide alternative	The system should provide alternative text descriptions for images used in the tagging	Medium	Murugaraj

Coding & solutioning

5.1 Feature 1

One key feature of autonomous tagging in Stack Overflow is the use of Natural Language Processing (NLP) techniques to extract relevant tags from the content of a question. This feature involves the following steps:Text analysis: The system analyzes the question's title, body, and any accompanying code snippets using NLP algorithms. This analysis helps in understanding the context and extracting important information.Entity recognition: The system identifies entities such as programming languages, frameworks, libraries, or specific technologies mentioned in the question. This can be done using techniques like named entity recognition or pattern matching.Keyword extraction: The system extracts keywords and key phrases from the question's text. These keywords help in determining the main topics or concepts discussed in the question**.**

```
app = Flask(_name_)

# Replace "<your API key>" with your actual IBM Watson API key

API_KEY = "1yZHLVnd1waigHLydqUB6NHzmAi_o9c0vO1bw7uTDUg3"

token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-
type:apikey'})

mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
mltoken}

@app.route('/')

def index():

    return render_template('index.html')

@app.route('/predict', methods=['POST'])

def predict():

   try:

        question = request.json['question']        payload_scoring = {"input_data":
[{"fields": ["question"], "values": [[question]]}]}
```

5.2 Feature 2

By employing NLP-based tag extraction, Stack Overflow can automate the initial tagging process, reducing manual effort and improving the consistency and accuracy of tag assignments. This feature enables faster question routing and improves the overall user experience by ensuring that questions are appropriately categorized and discoverable.

```
import joblib

import pandas as pd

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.preprocessing import MultiLabelBinarizer

import ast

# READ DATA

df = pd.read_csv('final-data.csv')

df1 = df.dropna().copy()  # Create a copy of the DataFrame

# Convert Tag column from string to list

df1['Tag'] = df1['Tag'].apply(lambda x: ast.literal_eval(x))

# Load the tagPredictorModel

tagPredictorModel = joblib.load('tagPredictor.pkl')

# Apply TfidfVectorizer

tfidf = TfidfVectorizer(analyzer='word', max_features=10000, ngram_range=(1, 3),
stop_words='english')

X = tfidf.fit_transform(df1['Body'].values.astype(str))

# Apply MultiLabelBinarizer on Tag column

multilabel = MultiLabelBinarizer()

df1['Tag'] = multilabel.fit_transform(df1['Tag'])
```

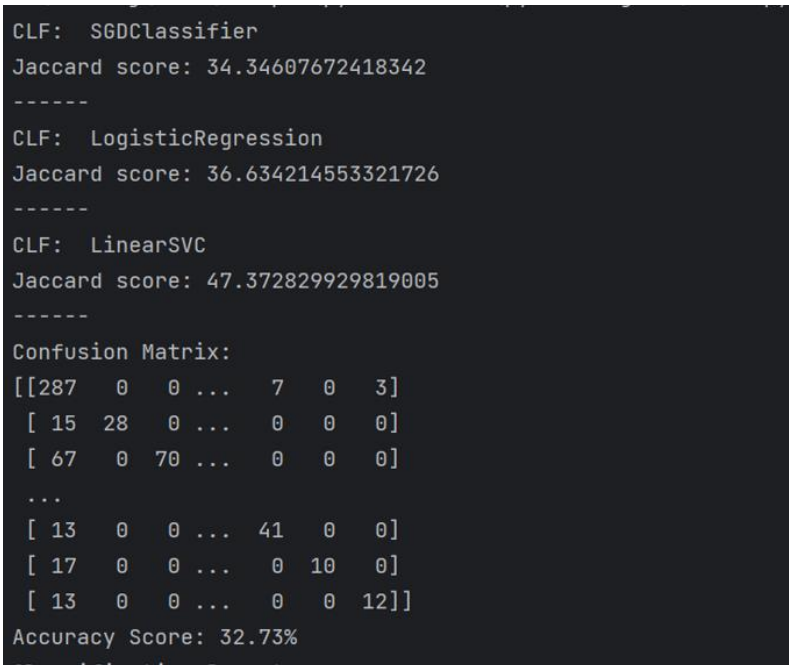
RESULT

6.1 Performance Metrics

performance metrics for a model, you typically need a set of labeled data (with known ground truth) to evaluate the model's predictions. The choice of performance metrics depends on the type of problem you are solving, such as classification, regression, or clustering. Here are some common performance metrics for different types of models

Classification Metrics:

- Accuracy: Accuracy measures how well the autonomous tagging system assigns the correct tags to questions. It is calculated by comparing the system's assigned tags with manually assigned or ground truth tags. The accuracy metric provides an overall measure of the system's tag assignment correctness.
- Precision: Precision focuses on the proportion of correctly assigned tags out of all the tags assigned by the autonomous tagging system. It measures the system's ability to avoid assigning irrelevant or incorrect tags to questions. A higher precision indicates a lower rate of false positives.
- Recall: Recall measures the proportion of correctly assigned tags out of all the relevant tags for a particular question. It assesses the system's ability to capture all the relevant tags and avoid missing important ones. A higher recall indicates a lower rate of false negatives.
- F1 score: The F1 score is the harmonic mean of precision and recall. It provides a balanced measure that considers both precision and recall. F1 score is often used when there is an imbalanced distribution of tags in the dataset.
- Top-N accuracy: Stack Overflow allows multiple tags to be assigned to a question. Top-N accuracy measures whether the system's assigned tags include at least one correct tag in the top N suggestions. This metric evaluates the system's ability to provide relevant tags within the top N predictions.
- Tag coverage: Tag coverage assesses the proportion of questions in the dataset for which the autonomous tagging system can assign at least one tag. A higher tag coverage indicates a broader range of questions that can be accurately tagged.
- Mean Average Precision (MAP): MAP evaluates the system's performance in ranking tags for a given question. It measures the average precision relevant tags, considering the position of correct tags in the ranked list of suggestions.



```
CLF:  SGDClassifier
Jaccard score: 34.34607672418342
------
CLF:  LogisticRegression
Jaccard score: 36.634214553321726
------
CLF:  LinearSVC
Jaccard score: 47.372829929819005
------
Confusion Matrix:
[[287   0   0 ...   7   0   3]
 [ 15  28   0 ...   0   0   0]
 [ 67   0  70 ...   0   0   0]
 ...
 [ 13   0   0 ...  41   0   0]
 [ 17   0   0 ...   0  10   0]
 [ 13   0   0 ...   0   0  12]]
Accuracy Score: 32.73%
```

ADVANTAGES & DISADVANTAGES

Advantages

- Increased efficiency: Autonomous tagging automates the process of assigning tags to Stack Overflow questions, saving time and effort for users, moderators, and community members. It eliminates the need for manual tagging and reduces human error in the process.
- Consistency and accuracy: Autonomous tagging helps ensure consistent and accurate tagging across the platform. By using predefined rules and machine learning algorithms, the system can assign tags consistently based on the content of the question, reducing inconsistencies that may arise from manual tagging.
- Improved search and discoverability: Accurate and relevant tags enable users to easily search for questions related to their specific interests or expertise. Autonomous tagging enhances the search experience by providing more accurate and comprehensive results.
- Facilitates question routing: Proper tagging helps route questions to the appropriate experts and community members who can provide relevant answers. This ensures that questions reach the right audience, increasing the chances of getting helpful responses.

Disadvantages

- Tagging errors: Autonomous tagging systems are not infallible and may occasionally assign incorrect or irrelevant tags to questions. The system's accuracy heavily relies on the quality of training data and the performance of the underlying algorithms. Inaccurate tags can lead to misclassification and hinder search and discovery.
- Difficulty in handling nuanced context: Certain questions may require a deep understanding of the context or domain-specific knowledge to assign accurate tags. Autonomous tagging systems may struggle with such nuanced scenarios, potentially leading to less precise tag assignments.
- Limited coverage: Autonomous tagging systems heavily rely on the available training data and predefined tag sets. If the system encounters tags that are not present in the training data or not part of the predefined set, it may struggle to assign appropriate tags, resulting in reduced coverage.
- Language and cultural biases: Autonomous tagging systems may inadvertently reflect biases present in the training data, leading to biased tag assignments. This can result in unequal representation or exclusion of certain topics, languages, or communities.
- Maintenance and adaptation: As programming languages, frameworks, and technologies evolve, the tag set needs to be regularly updated to reflect the latest trends. Ensuring the autonomous tagging system remains up-to-date requires ongoing maintenance and adaptation efforts

Conclusion

In conclusion, autonomous tagging of Stack Overflow questions brings several benefits to the platform and its users. By leveraging natural language processing and machine learning techniques, autonomous tagging automates the process of assigning relevant tags to questions. This improves efficiency, consistency, and accuracy in tagging, resulting in enhanced search and discoverability of questions. It also facilitates effective question routing, ensuring that questions reach the right experts for timely and helpful responses.However, there are limitations to autonomous tagging systems, including the potential for tagging errors, difficulty in handling nuanced context, limited coverage, and the presence of language and cultural biases. These factors should be considered and addressed to mitigate any negative impact on the tagging process.Overall, autonomous tagging is a valuable tool for managing the vast amount of questions on Stack Overflow and other similar platforms. It helps organize and categorize content, making it easier for users to find relevant information and connect with experts in their areas of interest. With continuous maintenance and improvement, autonomous tagging systems can play a significant role in enhancing the user experience and knowledge sharing within programming and development communities.

Future scope

The future scope of autonomous tagging of Stack Overflow questions holds several exciting possibilities. Here are some potential areas for further development and enhancement:

Advanced machine learning models: Continued advancements in machine learning and natural language processing techniques can lead to more sophisticated models for autonomous tagging. Deep learning approaches, such as convolutional neural networks (CNNs) or transformer-based models, can further improve the accuracy and performance of tag assignment.Context-aware tagging: Future systems can aim to capture more uanced context from the questions and their surrounding discussions. This can involve considering the question's temporal relevance, the expertise of users providing answers, or the evolving nature of programming languages and frameworks. Context-aware tagging can provide more precise and up-to-date tag assignments.User feedback integration: Incorporating user feedback into the autonomous tagging process can help refine the system over time. Users can provide feedback on the accuracy of assigned tags, allowing the system to learn from their input and adjust tag assignments accordingly. This iterative feedback loop can improve the system's performance and address tagging errors.Customizable tag sets: Allowing users to customize or personalize their tag preferences can enhance the user experience. Future systems can provide options for users to define their own tag sets or modify existing ones. This flexibility can cater to users' specific interests, making the tagging system more tailored to individual needs.Multilingual support: Stack Overflow has a global user base, and supporting multiple languages in autonomous tagging can broaden the platform's reach. Developing techniques for multilingual tag extraction and assignment can ensure that questions in different languages are appropriately tagged, improving accessibility and inclusivity.Collaboration with the community: Engaging the Stack Overflow community in the development and improvement of the autonomous tagging system can lead to better outcomes. Gathering insights and suggestions from moderators and experienced users can help refine the tagging algorithms and address specific challenges that arise in real-world scenarios.Ethical considerations: As autonomous tagging systems evolve, it is important to address ethical considerations, including bias detection and mitigation. Ensuring fair and unbiased tag assignments is crucial to maintain inclusivity and equal representation across various programming domains and user communities.

Appendix

10.1 Source code

To train the model

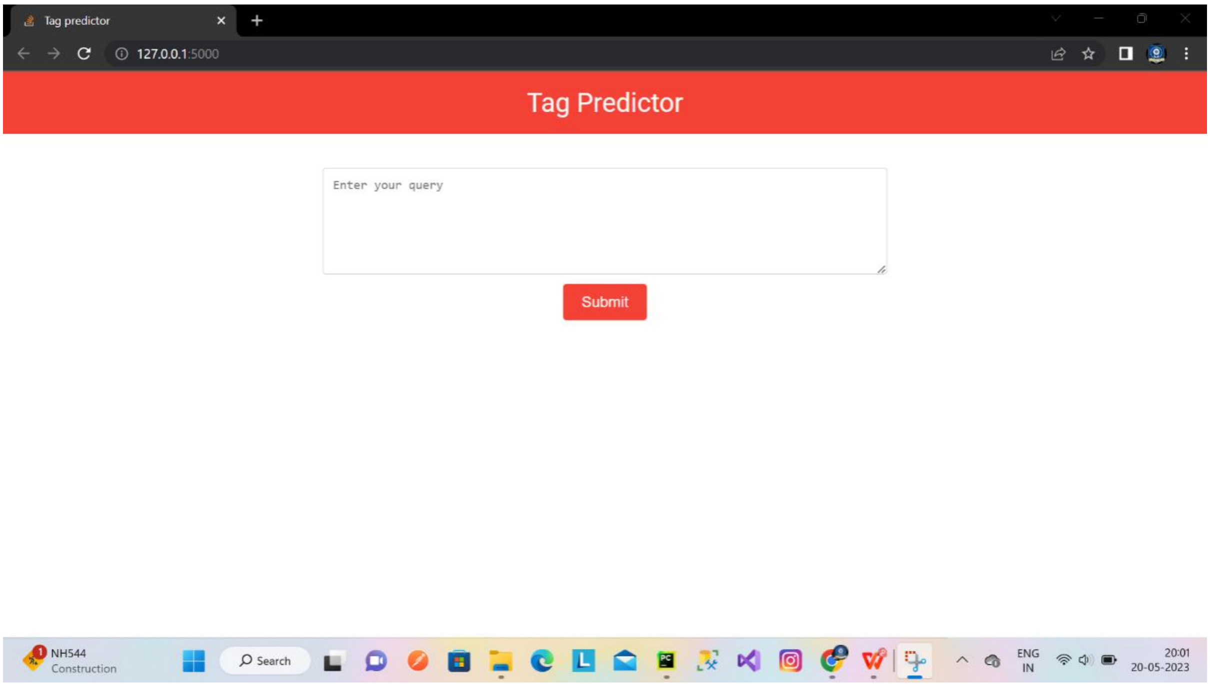
```
# IMPORTING LIBRARIES

import pandas as pd

import numpy as np

import ast

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.preprocessing import MultiLabelBinarizer

from sklearn.model_selection import train_test_split

from sklearn.linear_model import SGDClassifier, LogisticRegression

from sklearn.svm import LinearSVC
from sklearn.multiclass import OneVsRestClassifier
import joblib

# READ DATA
df = pd.read_csv('final-data.csv')
df.head()

# Tag columns is a string. We must convert it to a list.
df['Tag'] = df['Tag'].apply(lambda x: ast.literal_eval(x))
df.head()

# OBTAINING Y AS TARGET VARIABLE
y = df['Tag']

# CONVERT Y COLUMN TO CLASSES
multilabel = MultiLabelBinarizer()
y = multilabel.fit_transform(y)

# THE CLASSES
multilabel.classes_
pd.DataFrame(y, columns=multilabel.classes_)

# USING TF-IDF VECTORIZER
tfidf = TfidfVectorizer(analyzer='word', max_features=10000, ngram_range=(1, 3), stop_words='english')
X = tfidf.fit_transform(df['Body'].values.astype(str))
```

```
X.shape, y.shape


# SPLITTING DATA INTO TEST AND TRAIN SETS

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

tfidf.vocabulary_


# BUILD MODELS

sgd = SGDClassifier()

lr = LogisticRegression()

svc = LinearSVC()


def j_score(y_true, y_pred):

    # JACCARD SCORE IS USED TO CHECK THE ACCURACY OF A MULTILABEL CLASSIFICATION MODEL

    jaccard = np.minimum(y_true, y_pred).sum(axis=1) / np.maximum(y_true, y_pred).sum(axis=1)

    return jaccard.mean() * 100


def print_score(y_pred, clf):

    print("CLF: ", clf._class.name_)

    print("Jaccard score: {}".format(j_score(y_test, y_pred)))

    print("------")


for classifier in [sgd, lr, svc]:

    clf = OneVsRestClassifier(classifier)

    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_test)

    print_score(y_pred, classifier)


# EXPORTING MODEL

joblib_file = "tagPredictor.pkl"

joblib.dump(clf, joblib_file)


# Load from file

tagPredictorModel = joblib.load('tagPredictor.pkl')


def getTags(question):

    question = tfidf.transform(question)

    tags = multilabel.inverse_transform(tagPredictorModel.predict(question))

    print(tags)

from flask import Flask, render_template, jsonify, request

import requests


app = Flask(_name_)


# Replace "<your API key>" with your actual IBM Watson API key

API_KEY = "1yZHLVnd1waigHLydqUB6NHzmAi_o9c0vO1bw7uTDUg3"

token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})

mltoken = token_response.json()["access_token"]


header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
```

```
@app.route('/')
def index():
    return render_template('index.html')


@app.route('/predict', methods=['POST'])
def predict():
    try:
        question = request.json['question']
        payload_scoring = {"input_data": [{"fields": ["question"], "values": [[question]]}]}
        response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/95f044a0-b82d-42e6-9aaf-4760b2aa7b89/predictions?version=2023-05-20', json=payload_scoring, headers=header)
        tags = response_scoring.json()['predictions'][0]['values'][0]
        return jsonify(tags)
    except KeyError:
        error_message = "Question not found in the request."
        return jsonify(error=error_message), 400
    app.run(debug=True)
import joblib
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import MultiLabelBinarizer
import ast


# READ DATA
```

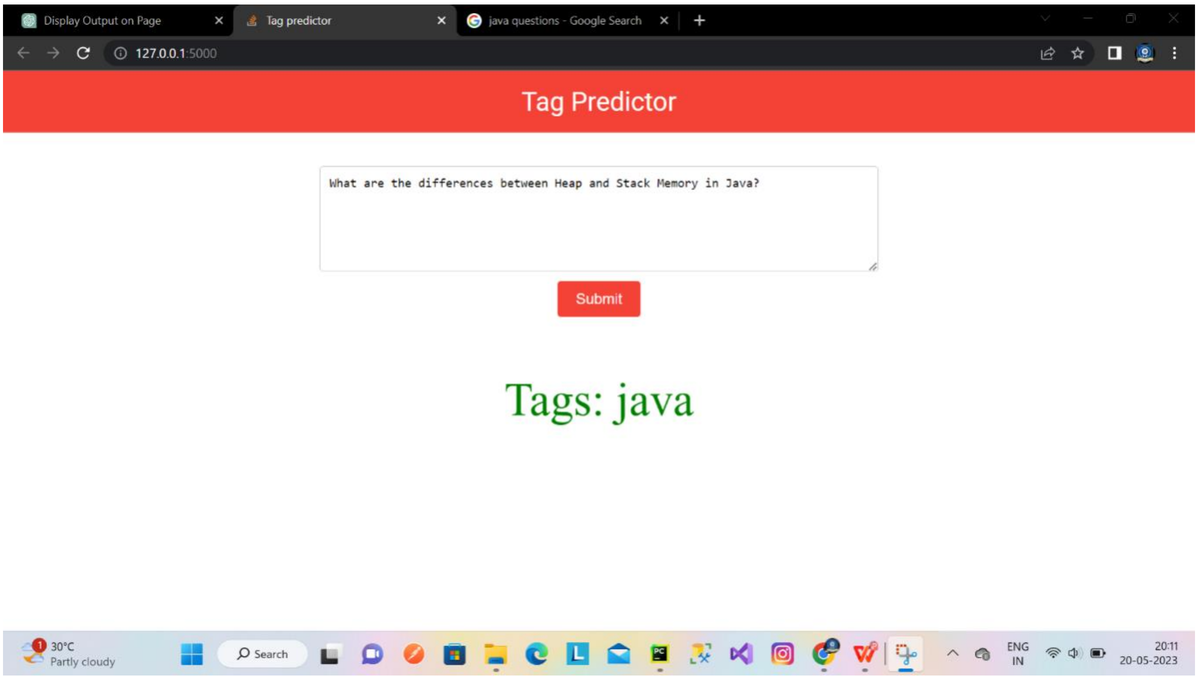
```
df = pd.read_csv('final-data.csv')

df1 = df.dropna().copy()  # Create a copy of the DataFrame



# Convert Tag column from string to list

df1['Tag'] = df1['Tag'].apply(lambda x: ast.literal_eval(x))



# Load the tagPredictorModel

tagPredictorModel = joblib.load('tagPredictor.pkl')



# Apply TfidfVectorizer

tfidf = TfidfVectorizer(analyzer='word', max_features=10000, ngram_range=(1, 3),
stop_words='english')

X = tfidf.fit_transform(df1['Body'].values.astype(str))



# Apply MultiLabelBinarizer on Tag column

multilabel = MultiLabelBinarizer()

df1['Tag'] = multilabel.fit_transform(df1['Tag'])



def getTags(question):

    question = tfidf.transform(question)

    tags = multilabel.inverse_transform(tagPredictorModel.predict(question))

    print(tags)

    return tags
```

OUTPUT:



Tag Predictor

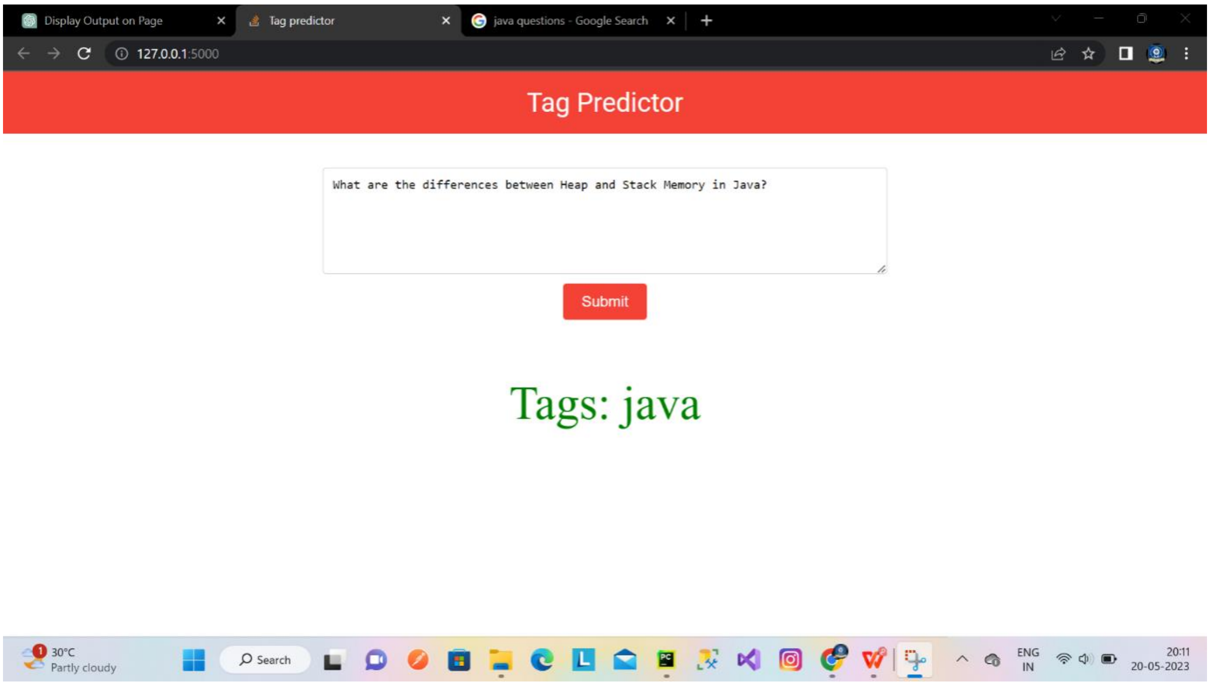
Enter your query



Tag Predictor

What are the differences between Heap and Stack Memory in Java?

Tags: java



Tag Predictor

What are the differences between Heap and Stack Memory in Java?

Tags: java

Github Video Link :

https://github.com/naanmudhalvan-SI/IBM--18431-1682492226

Project Demo:

https://drive.google.com/file/d/100aL9FDWusZ7v4p5fuzAddnLK4kYjcES/view?usp=sharing