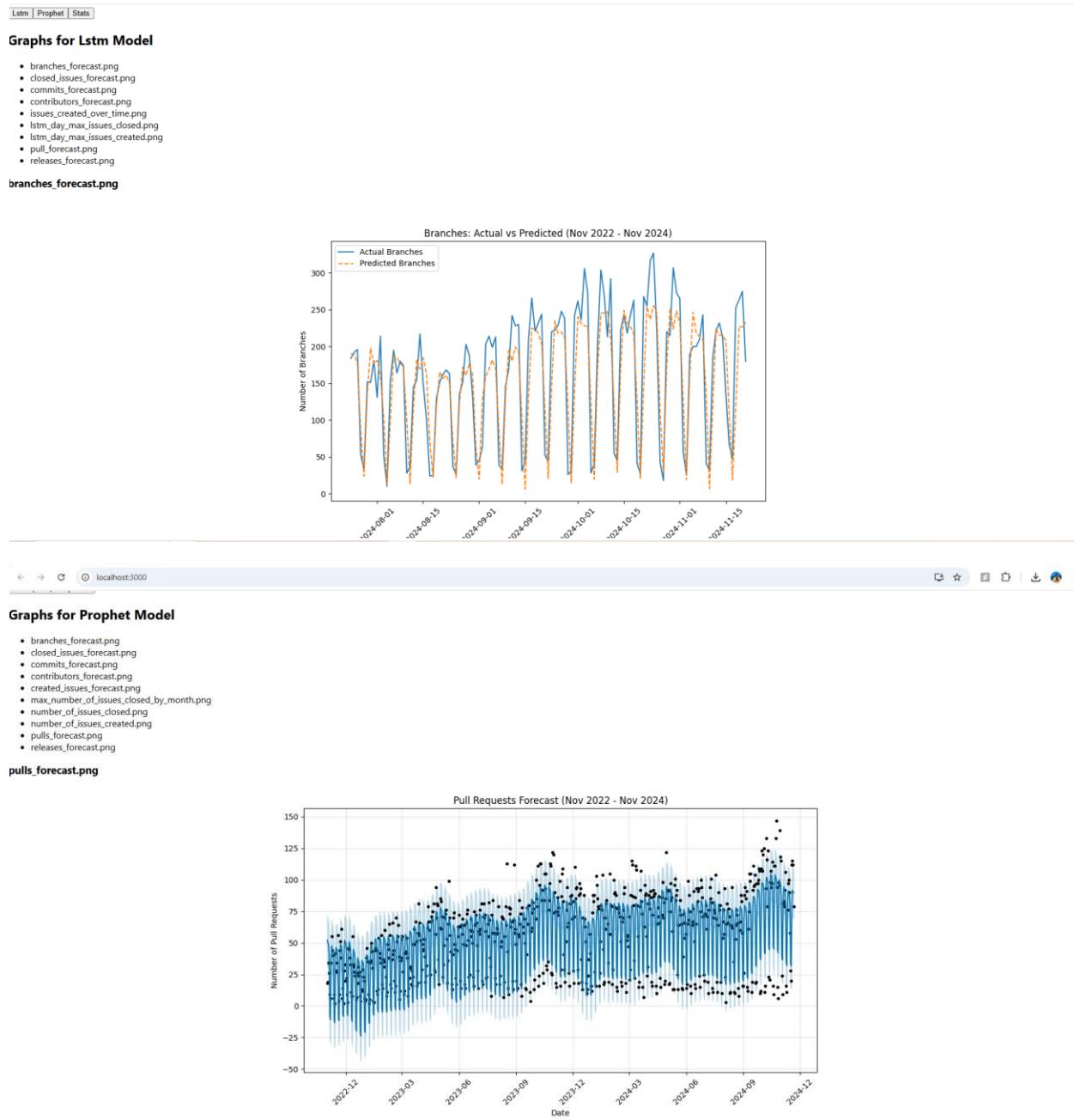


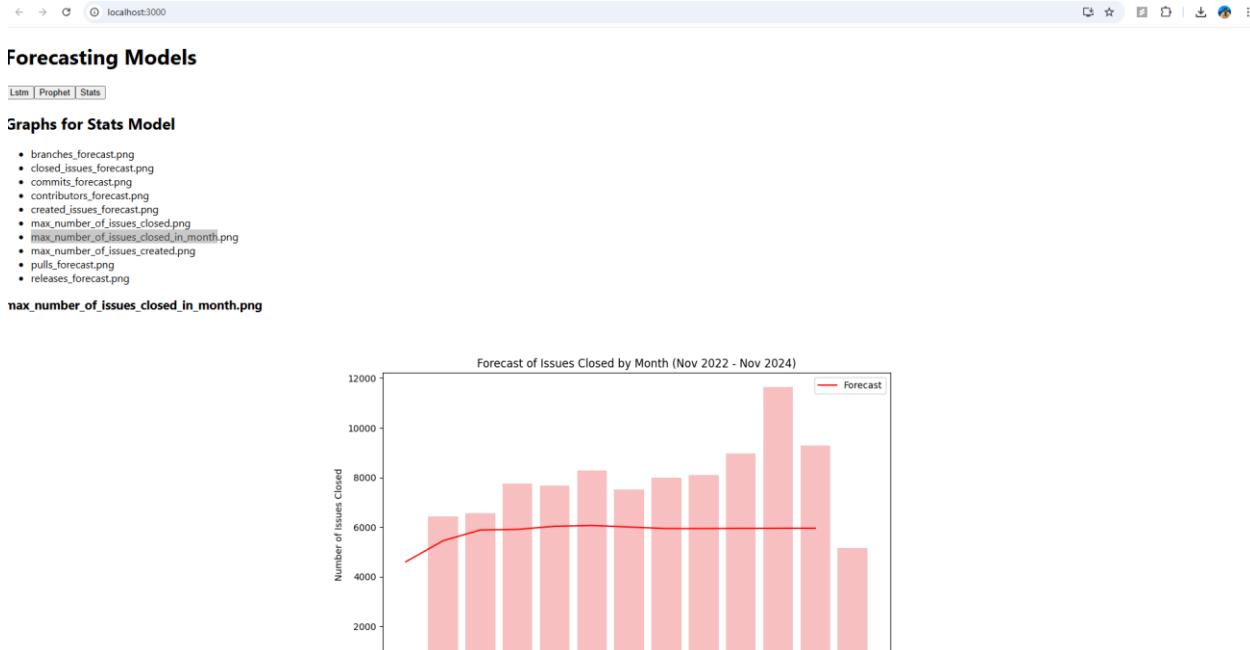
Name – Soham Sonar

CWID: A20541266

PART 1

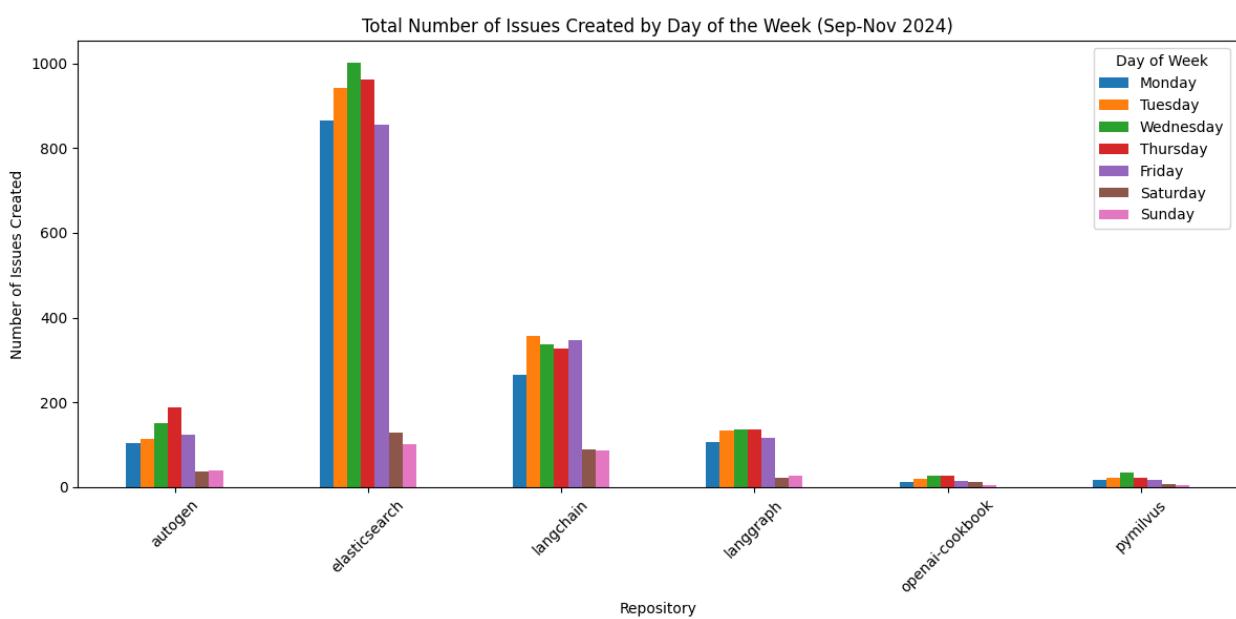
Frontend of our app for All the three – LSTM, Prophet, Stats Models and their graphs For all the repos.





Part 2

Requirement 1: Graph of Total Number of Issues by Day of the Week for each repo.



Requirement 2 -

Step 1 – Creating elasticsearch image and Push

```
PS C:\Users\soham> cd "E:\Masters at IIT\Masters_Sem_3\Software Project Management\Assignment_5"
PS E:\Masters at IIT\Masters_Sem_3\Software Project Management\Assignment_5> cd Github_Issues_ES_Docker_OpenAI
PS E:\Masters at IIT\Masters_Sem_3\Software Project Management\Assignment_5\Github_Issues_ES_Docker_OpenAI> cd Push_github_Issues
PS E:\Masters at IIT\Masters_Sem_3\Software Project Management\Assignment_5\Github_Issues_ES_Docker_OpenAI\Push_github_Issues> docker-compose up -d
error during connect: Get "http://222.222.222.222:5000/_labels": dial tcp 222.222.222.222:5000: connect: connection refused
PS E:\Masters at IIT\Masters_Sem_3\Software Project Management\Assignment_5\Github_Issues_ES_Docker_OpenAI\Push_github_Issues> docker-compose up -d
  ⚡  Rebuilding "push_github_issues_default" [1/2]
  ⚡  Creating network "push_github_issues_default" with the default driver
  ⚡  Creating container "push_github_issues-elasticsearch-1" started
  ⚡  Creating network "push_github_issues_default" with the default driver
PS E:\Masters at IIT\Masters_Sem_3\Software Project Management\Assignment_5\Github_Issues_ES_Docker_OpenAI\Push_github_Issues>
```

In push python file Correctly placing our repos and getting data of 2 months-



The screenshot shows a code editor with three separate code blocks, each labeled 'Python' in the top right corner. The first code block contains:

```
# Declare the owner and the repository
owners = ['langchain-ai', 'microsoft', 'openai', 'elastic', 'milvus-io']
repos = ['langchain', 'langgraph', 'autogen', 'openai-cookbook', 'elasticsearch', 'pymilvus']
```

The second code block contains:

```
page = 1
per_page = 100
from_date = (dt.date.today() - dt.timedelta(days=60)).isoformat() #The duration for which we need the issues can be changed here.
```

The third code block contains:

```
# Method that returns the base url
def fetch_url(owner, repo):
    return f"https://:{headers['Git_Username']}:{headers['access_token']}@api.github.com/repos/{owner}/{repo}"
```

Inserted data in elastic search

```

# Bulk indexing for githubissues

def dataframe_to_bulk_actions(df_Issues):
    for index, row in df_Issues.iterrows():
        yield {
            "_index": 'github_issues',
            "_source": {
                "_type": row['_type'],
                "_repo": row['_repo'],
                "_issueNumber": row['_issueNumber'],
                "_title": row['_title'],
                "_createdAt": row['_createdAt'],
                "_closedAt": row['_closedAt'],
                "_state": row['_state'],
                "_body": row['_body'],
                "GitHub_Issue_vector": row['GitHub_Issue_vector']
            }
        }

start = 0
end = len(df_Issues)
batch_size = 500

for batch_start in range(start, end, batch_size):
    batch_end = min(batch_start + batch_size, end)
    batch_dataframe = df_Issues.iloc[batch_start:batch_end]
    actions = list(dataframe_to_bulk_actions(df_Issues.iloc[start:end]))

    success, failed = helpers.bulk(es, actions)
    print(f"Inserted {success} records into Elasticsearch. Failed records: {failed}")

✓ 15.4s
Inserted 3910 records into Elasticsearch. Failed records: []

```

Push container running-

The screenshot shows the Docker Desktop interface. On the left, a sidebar menu includes 'Containers', 'Images', 'Volumes', 'Builds', 'Docker Scout', and 'Extensions'. The main area is titled 'Containers' with a 'Give feedback' link. It displays resource usage statistics: 'Container CPU usage' at 0.50% / 800% (8 CPUs available) and 'Container memory usage' at 4.04GB / 6.59GB. A 'Show charts' button is present. Below this, a search bar and a filter button ('Only show running containers') are shown. A table lists the running container, with one entry: Name: push_github_iss, Container ID: -, Image: -, Port(s): -, CPU (%): 0.41%, La: 36, Actions: a three-dot menu and a trash icon.

Committed and pushed the image:

```
PS E:\Masters at IIT\Masters_Sem_3\Software Project Management\Assignment_5\Github_Issues_ES_Docker_OpenAI\Push_github_Issues> docker commit 325228249a24 ssonar2/elasticsearch_vector:latest
325228249a24: Pushed
PS E:\Masters at IIT\Masters_Sem_3\Software Project Management\Assignment_5\Github_Issues_ES_Docker_OpenAI\Push_github_Issues> docker push ssonar2/elasticsearch_vector:latest
The push refers to repository [docker.io/ssonar2/elasticsearch_vector]
1e8bb489bb5a: Pushed
f90853ba4ea4: Layer already exists
cf37f101717: Layer already exists
c34ee2ffa242: Layer already exists
7536ca9d0933: Layer already exists
76753e22089e: Layer already exists
1840aa78e769: Layer already exists
5f70bf18a086: Layer already exists
ba4dd2005fa: Layer already exists
9293d92aa965: Layer already exists
3a03f09d2129: Layer already exists
latest: digest: sha256:1b39a63363e21e91a1fe4941dfa5287da54cc780cafela820e32e6f6d5c693f5 size: 2623
PS E:\Masters at IIT\Masters_Sem_3\Software Project Management\Assignment_5\Github_Issues_ES_Docker_OpenAI\Push_github_Issues> ■
```

Successfully pushed and deployed image to docker hub

```
new build details: docker-desktop://dashboard/build/unruffled_boyd/unruffled_boyd0/pn4w7k0pac7mjtnz1c3kb8g6
+] Building 7.2s (9/9) FINISHED
> [internal] load build definition from Dockerfile
--> => transferring dockerfile: 77B
[+] [linux/amd64 internal] load metadata for docker.io/ssonar2/elasticsearch_vector:latest
[+] [linux/amd64 internal] load metadata for docker.io/ssonar2/elasticsearch_vector:latest
--> [auth] ssonar2/elasticsearch_vector:pull token for registry-1.docker.io
[+] [internal] load .dockerignore
--> => transferring context: 2B
[+] [linux/amd64 1/1] FROM docker.io/ssonar2/elasticsearch_vector:latest@sha256:1b39a63363e21e91a1fe4941dfa5287da54cc780cafe1a820e32e6f6d5c6935f
--> => resolve docker.io:ssonar2/elasticsearch_vector:latest@sha256:1b39a63363e21e91a1fe4941dfa5287da54cc780cafe1a820e32e6f6d5c6935f
--> => resolve docker.io:ssonar2/elasticsearch_vector:latest@sha256:1b39a63363e21e91a1fe4941dfa5287da54cc780cafe1a820e32e6f6d5c6935f
--> => resolve docker.io:ssonar2/elasticsearch_vector:latest@sha256:1b39a63363e21e91a1fe4941dfa5287da54cc780cafe1a820e32e6f6d5c6935f
--> => exporting to image
--> => exporting layers
--> => exporting manifest sha256:42597a4ccbbcc0cf9a9886de3b4705d78da9017e7a4be207432a5ee6030336
--> => exporting config sha256:3d779ca07471a94867d323b2cf0eaaf1324c838e174a44623ea551c2619358d0
--> => exporting attestation manifest sha256:b5c3dd6b099b251767ae0387c56cc62fc3c2ee31a8f4ef2812d7fc7b45f14a
--> => exporting manifest sha256:116ba095c1b3f3427d0b64ba8e07853902426d870ed1c13e188c352d8173c
--> => exporting config sha256:49245fd753fb41fc77d0605d2d9363537c77f76538ccaf1dddebfc280
--> => exporting attestation manifest sha256:a5dd32746ca4c54a13ac775a6b29763aa5bc3cd85826572c9f44161af18b8eb
--> => exporting manifest list sha256:aaab566183e7f6731f31569c769a7fe7585d37a72287a246ba9760fc7a119a67
--> => pushing layers
--> => pushing manifest for docker.io:ssonar2/elasticsearch_vector:multi@sha256:aaab566183e7f6731f31569c769a7fe7585d37a72287a246ba9760fc7a119a67
--> [auth] ssonar2/elasticsearch_vector:pull,push token for registry-1.docker.io

new build details: docker-desktop://dashboard/build/unruffled_boyd/unruffled_boyd0/pn4w7k0pac7mjtnz1c3kb8g6
S:\Masters at IIT\Masters_Sem_3\Software Project Management\Assignment_5\Github_Issues_ES_Docker_OpenAI\Push_github_Issues>
```

Step 2: Pulling our elastic search image

```
E:\Masters at IIT\Masters_Sem_3\Software Project Management\Assignment_5\Github_Issues_ES_Docker_OpenAI> cd Pull_github_Issues
E:\Masters at IIT\Masters_Sem_3\Software Project Management\Assignment_5\Github_Issues_ES_Docker_OpenAI> docker-compose up -d
[...]
Network pull_github_issues_default      Created
Container pull_github_issues-elasticsearch-1 Started
E:\Masters at IIT\Masters_Sem_3\Software Project Management\Assignment_5\Github_Issues_ES_Docker_OpenAI> Pull_github_Issues
```

New More Docker. Easy Access. New Streamlined Plan

dockerhub Explore Repositories Organizations Usage Search Docker Hub

ssonar2 / [Repositories](#) / [elasticsearch_vector](#) / [General](#)

General Tags Builds Collaborators Webhooks Settings

ssonar2/elasticsearch_vector

Last pushed 1 minute ago

This repository does not have a description

This repository does not have a category

Tags

This repository contains 2 tag(s).

Tag	OS	Type	Pulled	Pushed
multi		Image	a few seconds ago	a few seconds ago
latest		Image	7 minutes ago	7 minutes ago

[See all](#)

Successfully pulled our elastic search image for all repositories

```
# To search documents

#Import Elasticsearch and helpers from .elasticsearch

from elasticsearch import Elasticsearch,helpers
import json

es = Elasticsearch(['http://localhost:9200'])
es.ping()  #connection testing
res = es.search(index="github_issues", body={"query": {"match_all": {}}})

# Extract the total hit count
print("Got %d Hits:" % res['hits']['total']['value'])

# Extract hits from the response
hits = res.get('hits', {})

# Pretty print the hits
print(json.dumps(hits, indent=2))

[1] ✓ 0.2s Python

Got 3911 Hits:
{
  "total": {
    "value": 3911,
    "relation": "eq"
  },
  "max_score": 1.0,
  "hits": [
    {
      "_index": "github_issues",
      "_id": "AmLQUZMBKaR7I-arfsTA",
      "score": 1.0,
```

Got output and message for repos which do not have any similar issues recently-

```
-----  
Top 5 similar issues for openai/openai-cookbook:  
-----  
No recent similar issues from last 2 months found for openai/openai-cookbook  
-----  
Top 5 similar issues for elastic/elasticsearch:  
-----  


```
{
 "_score": 1.0,
 "_type": "issue",
 "_repo": "elasticsearch",
 "_issueNumber": "116131",
 "_title": "[8.16] Resolve pipelines from template if lazy rollover write (#116031)",
 "_createdAt": "2024-11-02T03:56:02Z",
 "_closedAt": "2024-11-02T17:12:07Z",
 "_body": "Backports the following commits to 8.16:\n - Resolve pipelines from template if lazy rollover write (#116031)"
}
{
 "_score": 0.96739626,
 "_type": "issue",
 "_repo": "elasticsearch",
 "_issueNumber": "116131",
 "_title": "[8.16] Resolve pipelines from template if lazy rollover write (#116031)",
 "_createdAt": "2024-11-02T03:56:02Z",
 "_closedAt": "2024-11-02T17:12:07Z",
 "_body": "Backports the following commits to 8.16:\n - Resolve pipelines from template if lazy rollover write (#116031)"
}
{
 "_score": 0.96739626,
 "_type": "issue",
 "_repo": "elasticsearch",
 "_issueNumber": "116131",
 "_title": "[8.16] Resolve pipelines from template if lazy rollover write (#116031)",
 "_createdAt": "2024-11-02T03:56:02Z",
 "_closedAt": "2024-11-02T17:12:07Z",
 "_body": "Backports the following commits to 8.16:\n - Resolve pipelines from template if lazy rollover write (#116031)"
}
{
 "_score": 0.96739626,
 "_type": "issue",
 "_repo": "elasticsearch",
 "_issueNumber": "116131",
 "_title": "[8.16] Resolve pipelines from template if lazy rollover write (#116031)",
 "_createdAt": "2024-11-02T03:56:02Z",
 "_closedAt": "2024-11-02T17:12:07Z",
 "_body": "Backports the following commits to 8.16:\n - Resolve pipelines from template if lazy rollover write (#116031)"
}
{
 "_score": 0.96739626,
 "_type": "issue",
 "_repo": "elasticsearch",
 "_issueNumber": "116131",
 "_title": "[8.16] Resolve pipelines from template if lazy rollover write (#116031)",
 "_createdAt": "2024-11-02T03:56:02Z",
 "_closedAt": "2024-11-02T17:12:07Z",
 "_body": "Backports the following commits to 8.16:\n - Resolve pipelines from template if lazy rollover write (#116031)"
}
```


```

```
...
Name: Soham Sonar
CWID: A20541266
...
import warnings
warnings.filterwarnings('ignore')

import aiohttp
import asyncio
import nest_asyncio
from concurrent.futures import ThreadPoolExecutor
from datetime import datetime, timedelta
import pandas as pd

# Allow nested event loops
nest_asyncio.apply()

# GitHub Personal Access Token
GITHUB_TOKEN = "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"
HEADERS = {"Authorization": f"token {GITHUB_TOKEN}"}

# List of repositories
repositories = [
    ("langchain-ai", "langchain"),
    ("langchain-ai", "langgraph"),
    ("microsoft", "autogen"),
    ("openai", "openai-cookbook"),
    ("elastic", "elasticsearch"),
    ("milvus-io", "pymilvus"),
]
# Fetch data for a single repository
async def fetch_repo_data(session, owner, repo):
    # Fetch all repository details concurrently
    tasks = [
        get_repo_metadata(session, owner, repo),
        get_issues_details(session, owner, repo),
        get_commits_details(session, owner, repo),
        get_pull_requests(session, owner, repo),
        get_releases(session, owner, repo),
        get_contributors(session, owner, repo),
        get_branches(session, owner, repo),
    ]
    (
        metadata,
        issues,
        commits,
        pull_requests,
        releases,
        contributors,
```

```

        branches,
    ) = await asyncio.gather(*tasks)

    # Print counts only
    print(f"Repository: {repo}")
    print(f" Stars: {metadata['stars']}, Forks: {metadata['forks']} ")
    print(f" Issues: {len(issues)}, Commits: {len(commits)} ")
    print(f" Pull Requests: {len(pull_requests)}, Releases: "
{len(releases)} )
    print(f" Contributors: {len(contributors)}, Branches: "
{len(branches)} )

    return {
        "repository": repo,
        "stars": metadata["stars"],
        "forks": metadata["forks"],
        "issues": issues,
        "commits": commits,
        "pull_requests": pull_requests,
        "releases": releases,
        "contributors": contributors,
        "branches": branches,
    }

# Fetch all repositories using ThreadPoolExecutor
def fetch_all_repositories():
    async def fetch_all():
        async with aiohttp.ClientSession(headers=HEADERS) as session:
            tasks = [fetch_repo_data(session, owner, repo) for owner,
repo in repositories]
            return await asyncio.gather(*tasks)

    loop = asyncio.get_event_loop()
    return loop.run_until_complete(fetch_all())

# Asynchronous fetch function with pagination
async def fetch(session, url, params):
    all_data = []
    page = 1
    while True:
        params["page"] = page
        async with session.get(url, params=params) as response:
            if response.status == 403:
                print(f"Error fetching {url}: Rate limit exceeded.")
                break
            if response.status != 200:
                print(f"Error fetching {url}: {response.status}")
                break
            data = await response.json()
            if not data:

```

```

        break
    all_data.extend(data)
    page += 1
return all_data

# Functions to fetch specific data
async def get_repo_metadata(session, owner, repo):
    url = f"https://api.github.com/repos/{owner}/{repo}"
    async with session.get(url) as response:
        repo_info = await response.json()
        stars = repo_info.get("stargazers_count", 0)
        forks = repo_info.get("forks_count", 0)
        return {"stars": stars, "forks": forks}

async def get_issues_details(session, owner, repo, days=730):
    url = f"https://api.github.com/repos/{owner}/{repo}/issues"
    since = (datetime.now() - timedelta(days=days)).isoformat()
    params = {"state": "all", "since": since, "per_page": 100}
    return await fetch(session, url, params)

async def get_commits_details(session, owner, repo, days=730):
    url = f"https://api.github.com/repos/{owner}/{repo}/commits"
    since = (datetime.now() - timedelta(days=days)).isoformat()
    params = {"since": since, "per_page": 100}
    return await fetch(session, url, params)

async def get_pull_requests(session, owner, repo, days=730):
    url = f"https://api.github.com/repos/{owner}/{repo}/pulls"
    since = (datetime.now() - timedelta(days=days)).isoformat()
    params = {"state": "all", "since": since, "per_page": 100}
    return await fetch(session, url, params)

async def get_releases(session, owner, repo):
    url = f"https://api.github.com/repos/{owner}/{repo}/releases"
    params = {"per_page": 100}
    return await fetch(session, url, params)

async def get_contributors(session, owner, repo):
    url = f"https://api.github.com/repos/{owner}/{repo}/contributors"
    params = {"per_page": 100}
    return await fetch(session, url, params)

async def get_branches(session, owner, repo):
    url = f"https://api.github.com/repos/{owner}/{repo}/branches"
    params = {"per_page": 100}
    return await fetch(session, url, params)

# Main program
try:
    print("Fetching data from GitHub...")

```

```
with ThreadPoolExecutor() as executor:
    all_data = executor.submit(fetch_all_repositories).result()

# Prepare detailed data
detailed_data = []
summary_data = []
for repo in all_data:
    # Collect detailed data
    for issue in repo["issues"]:
        detailed_data.append({
            "repository": repo["repository"],
            "type": "issue",
            "created_at": issue.get("created_at"),
            "closed_at": issue.get("closed_at"),
            "state": issue.get("state"),
        })
    for commit in repo["commits"]:
        detailed_data.append({
            "repository": repo["repository"],
            "type": "commit",
            "commit_date": commit.get("commit", {}).get("author", {}).get("date"),
            "author_name": commit.get("commit", {}).get("author", {}).get("name"),
            "message": commit.get("commit", {}).get("message"),
        })
    for pull_request in repo["pull_requests"]:
        detailed_data.append({
            "repository": repo["repository"],
            "type": "pull_request",
            "created_at": pull_request.get("created_at"),
            "closed_at": pull_request.get("closed_at"),
            "state": pull_request.get("state"),
        })
    for release in repo["releases"]:
        detailed_data.append({
            "repository": repo["repository"],
            "type": "release",
            "published_at": release.get("published_at"),
            "name": release.get("name"),
        })
    for contributor in repo["contributors"]:
        detailed_data.append({
            "repository": repo["repository"],
            "type": "contributor",
            "login": contributor.get("login"),
            "contributions": contributor.get("contributions"),
        })
    for branch in repo["branches"]:
```

```

        detailed_data.append({
            "repository": repo["repository"],
            "type": "branch",
            "branch_name": branch.get("name"),
        })

    # Collect summary data
    summary_data.append({
        "repository": repo["repository"],
        "stars": repo["stars"],
        "forks": repo["forks"],
        "total_issues": len(repo["issues"]),
        "total_commits": len(repo["commits"]),
        "total_pull_requests": len(repo["pull_requests"]),
        "total_releases": len(repo["releases"]),
        "total_contributors": len(repo["contributors"]),
        "total_branches": len(repo["branches"]),
    })

    # Create DataFrames
    df_detailed = pd.DataFrame(detailed_data)
    df_summary = pd.DataFrame(summary_data)

    # Save DataFrames to CSV
    df_detailed.to_csv("github_all_data.csv", index=False)
    df_summary.to_csv("github_only_summary_data.csv", index=False)

    print("Detailed and Summary data saved.")

except Exception as e:
    print(f"An error occurred: {e}")

Fetching data from GitHub...
Repository: openai-cookbook
Stars: 59891, Forks: 9554
Issues: 1364, Commits: 897
Pull Requests: 921, Releases: 0
Contributors: 250, Branches: 43
Repository: langgraph
Stars: 6769, Forks: 1094
Issues: 1899, Commits: 3068
Pull Requests: 1583, Releases: 146
Contributors: 88, Branches: 81
Repository: pymilvus
Stars: 1031, Forks: 329
Issues: 1218, Commits: 413
Pull Requests: 1477, Releases: 62
Contributors: 101, Branches: 14
Repository: autogen
Stars: 34609, Forks: 5002

```

```

Issues: 3935, Commits: 2241
Pull Requests: 1978, Releases: 61
Contributors: 372, Branches: 110
Repository: langchain
Stars: 95153, Forks: 15427
Issues: 23697, Commits: 11839
Pull Requests: 15963, Releases: 699
Contributors: 481, Branches: 843
Repository: elasticsearch
Stars: 1295, Forks: 24868
Issues: 31231, Commits: 14666
Pull Requests: 79817, Releases: 160
Contributors: 348, Branches: 422
Detailed and Summary data saved.

import pandas as pd
import matplotlib.pyplot as plt

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter only issue data
df_issues = df_detailed[df_detailed["type"] == "issue"]

# Convert created_at column to datetime
df_issues["created_at"] = pd.to_datetime(df_issues["created_at"])

# Filter data from 2022-11-01 to 2024-11-30
start_date = "2022-11-01"
end_date = "2024-11-30"
df_issues = df_issues[(df_issues["created_at"] >= start_date) &
(df_issues["created_at"] <= end_date)]

# Set created_at as index for easier resampling
df_issues.set_index("created_at", inplace=True)

# Resample data quarterly for each repository
quarterly_issues = (
    df_issues.groupby("repository")
    .resample("Q") # Group by quarter
    .size()
    .reset_index(name="issue_count") # Rename column for counts
)

# Create a pivot table for better plotting
pivot_data = quarterly_issues.pivot(index="created_at",
columns="repository", values="issue_count")

# Plot the line chart
plt.figure(figsize=(12, 8))

```

```
pivot_data.plot(kind="line", marker="o", figsize=(14, 8))

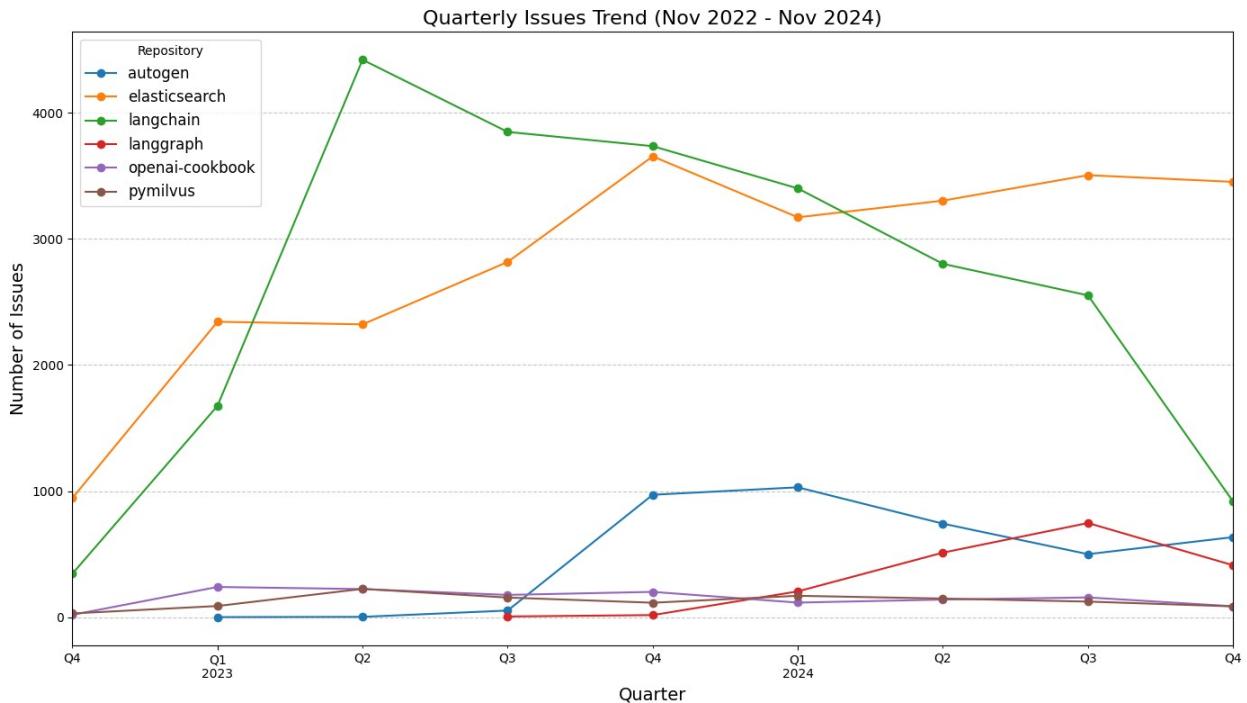
plt.title("Quarterly Issues Trend (Nov 2022 - Nov 2024)", fontsize=16)
plt.xlabel("Quarter", fontsize=14)
plt.ylabel("Number of Issues", fontsize=14)
plt.legend(title="Repository", fontsize=12)
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.tight_layout()

# Save or show the plot
plt.savefig("quarterly_issues_trend_filtered.png")
plt.show()

C:\Users\soham\AppData\Local\Temp\ipykernel_18460\3743448190.py:5:
DtypeWarning: Columns (5,6,7,8,9,10,12) have mixed types. Specify
dtype option on import or set low_memory=False.
    df_detailed = pd.read_csv("github_all_data.csv")
C:\Users\soham\AppData\Local\Temp\ipykernel_18460\3743448190.py:11:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
    df_issues["created_at"] = pd.to_datetime(df_issues["created_at"])
C:\Users\soham\AppData\Local\Temp\ipykernel_18460\3743448190.py:24:
FutureWarning: 'Q' is deprecated and will be removed in a future
version, please use 'QE' instead.
    .resample("Q") # Group by quarter

<Figure size 1200x800 with 0 Axes>
```



```
#Add your code for requirement 3 in this cell
```

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter only issue data
df_issues = df_detailed[df_detailed["type"] == "issue"]

# Convert created_at column to datetime
df_issues["created_at"] = pd.to_datetime(df_issues["created_at"])

# Filter data from 2022-11-01 to 2024-11-20
start_date = "2022-11-01"
end_date = "2024-11-20"
df_issues = df_issues[(df_issues["created_at"] >= start_date) &
                      (df_issues["created_at"] <= end_date)]

# Set created_at as index for easier resampling
df_issues.set_index("created_at", inplace=True)

# Resample data monthly for each repository
monthly_issues = (
    df_issues.groupby("repository")
    .resample("M") # Group by month
    .size()
```

```
.reset_index(name="issue_count") # Rename column for counts
)

# Create a pivot table for better plotting
pivot_data = monthly_issues.pivot(index="created_at",
columns="repository", values="issue_count")

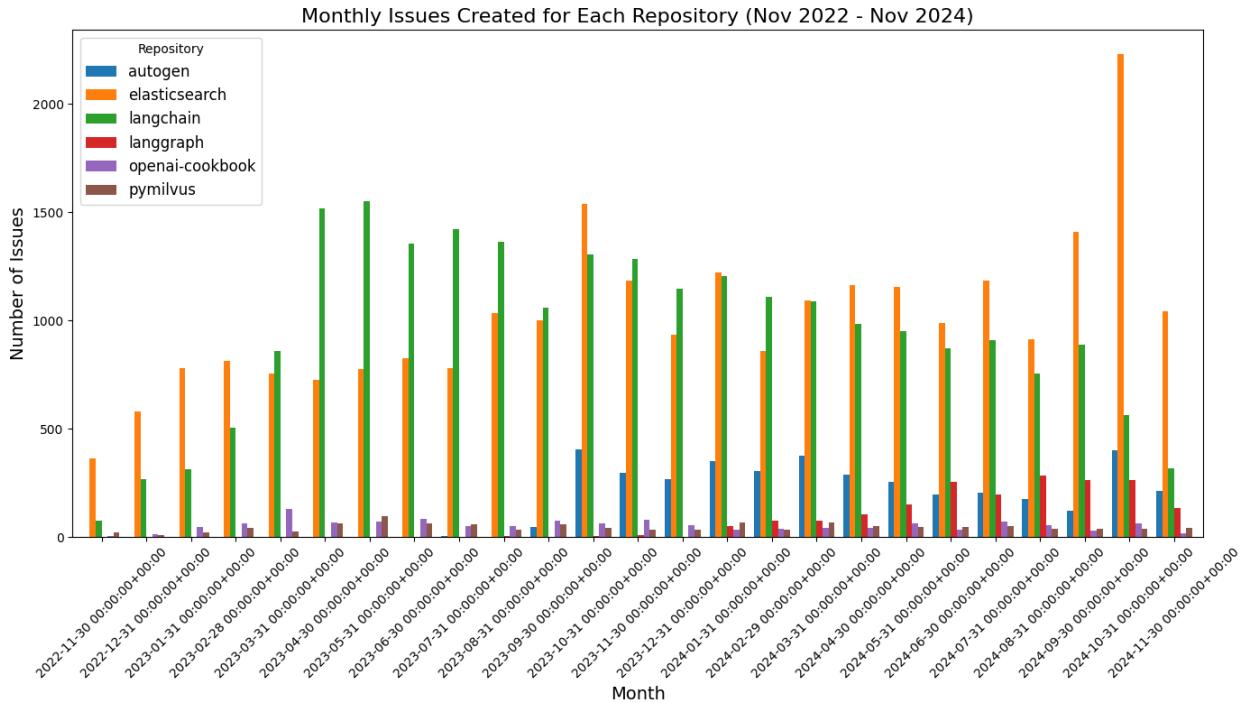
# Plot the bar chart
pivot_data.plot(kind="bar", figsize=(14, 8), width=0.8)

plt.title("Monthly Issues Created for Each Repository (Nov 2022 - Nov 2024)", fontsize=16)
plt.xlabel("Month", fontsize=14)
plt.ylabel("Number of Issues", fontsize=14)
plt.legend(title="Repository", fontsize=12)
plt.xticks(rotation=45)
plt.tight_layout()

# Save or show the plot
plt.savefig("monthly_issues_bar_chart.png")
plt.show()

C:\Users\soham\AppData\Local\Temp\ipykernel_18460\2105572857.py:7:
DtypeWarning: Columns (5,6,7,8,9,10,12) have mixed types. Specify
dtype option on import or set low_memory=False.
    df_detailed = pd.read_csv("github_all_data.csv")
C:\Users\soham\AppData\Local\Temp\ipykernel_18460\2105572857.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
    df_issues["created_at"] = pd.to_datetime(df_issues["created_at"])
C:\Users\soham\AppData\Local\Temp\ipykernel_18460\2105572857.py:26:
FutureWarning: 'M' is deprecated and will be removed in a future
version, please use 'ME' instead.
    .resample("M") # Group by month
```



```
#Add your code for requirement 4 in this cell
```

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the summary data
df_summary = pd.read_csv("github_only_summary_data.csv")

# Extract repository names and star counts
repo_stars = df_summary[["repository", "stars"]]

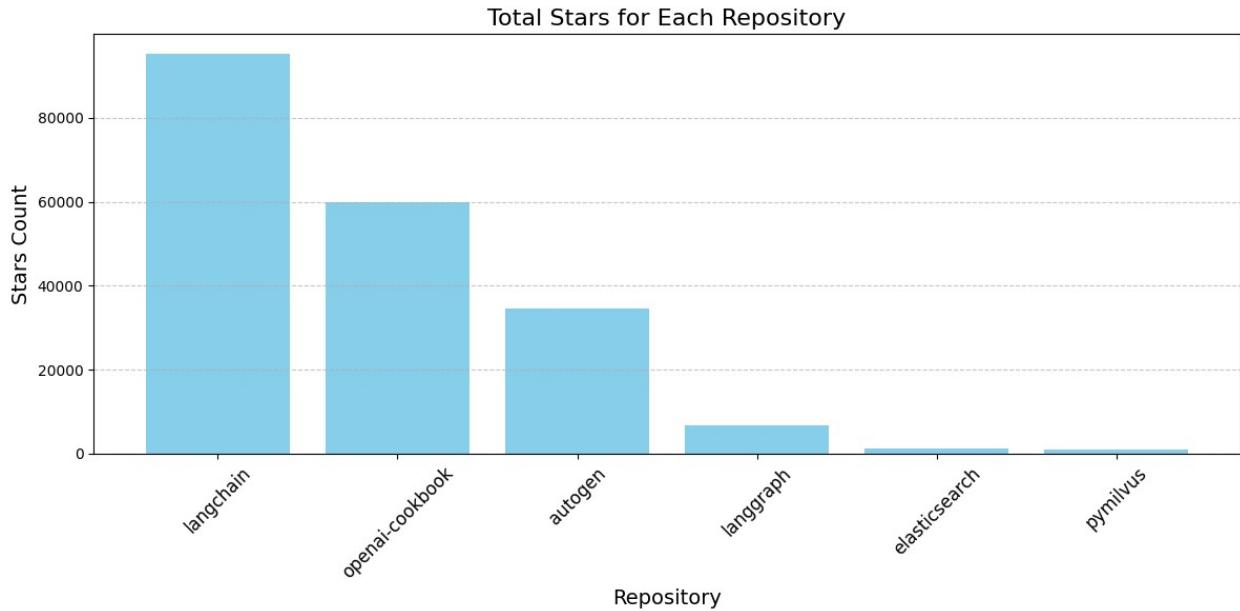
# Sort repositories by star count for better visualization
repo_stars = repo_stars.sort_values(by="stars", ascending=False)

# Plot the bar chart
plt.figure(figsize=(12, 6))
plt.bar(repo_stars["repository"], repo_stars["stars"],
color="skyblue")

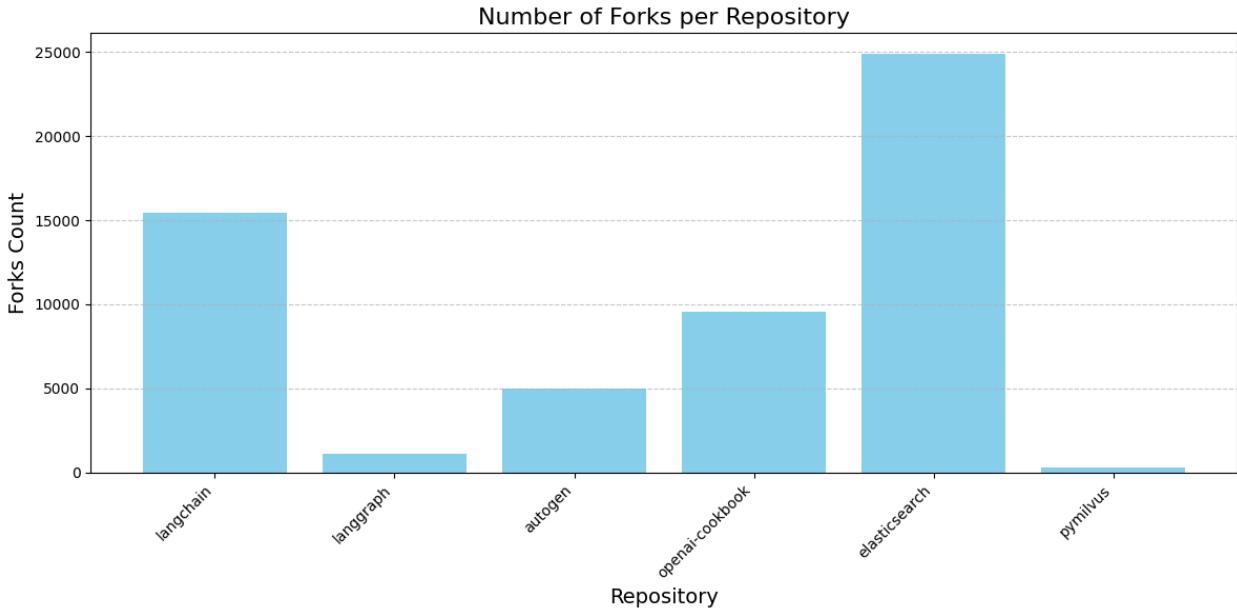
# Add titles and labels
plt.title("Total Stars for Each Repository", fontsize=16)
plt.xlabel("Repository", fontsize=14)
plt.ylabel("Stars Count", fontsize=14)
plt.xticks(rotation=45, fontsize=12)
plt.grid(axis="y", linestyle="--", alpha=0.7)

# Save or show the plot
plt.savefig("repository_stars_bar_chart.png")
```

```
plt.tight_layout()  
plt.show()
```



```
#Add your code for requirement 5 in this cell  
  
import pandas as pd  
import matplotlib.pyplot as plt  
  
# Load the summary data that contains the forks count  
df_summary = pd.read_csv("github_only_summary_data.csv")  
  
# Plotting a bar chart for the forks count  
plt.figure(figsize=(12, 6))  
plt.bar(df_summary['repository'], df_summary['forks'],  
color='skyblue')  
  
# Add titles and labels  
plt.title("Number of Forks per Repository", fontsize=16)  
plt.xlabel("Repository", fontsize=14)  
plt.ylabel("Forks Count", fontsize=14)  
plt.xticks(rotation=45, ha="right", fontsize=10)  
plt.grid(axis="y", linestyle="--", alpha=0.7)  
  
# Save or show the plot  
plt.tight_layout()  
plt.savefig("forks_per_repository_bar_chart.png")  
plt.show()
```



```
#Add your code for requirement 6 in this cell
```

```
import pandas as pd
import matplotlib.pyplot as plt
import datetime as dt

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter data for issues only and within the specified date range
df_issues = df_detailed[df_detailed["type"] == "issue"]
df_issues["closed_at"] = pd.to_datetime(df_issues["closed_at"],
errors="coerce")
df_issues = df_issues[(df_issues["closed_at"] >= "2022-11-01") &
(df_issues["closed_at"] <= "2024-11-30")]

# Add a week number for grouping
df_issues["week"] =
df_issues["closed_at"].dt.to_period("W").apply(lambda r: r.start_time)

# Group by repository and week, then count issues closed
weekly_closed_issues = df_issues.groupby(["repository",
"week"]).size().reset_index(name="closed_issues")

# Pivot the data for plotting
weekly_pivot = weekly_closed_issues.pivot(index="week",
columns="repository", values="closed_issues").fillna(0)

# Plot the bar chart
plt.figure(figsize=(14, 8))
```

```

weekly_pivot.plot(kind="bar", stacked=True, figsize=(16, 8),
width=1.0, colormap="tab10")

# Add titles and labels
plt.title("Weekly Closed Issues for Each Repository (past 2 years)",
fontsize=16)
plt.xlabel("Week", fontsize=14)
plt.ylabel("Closed Issues Count", fontsize=14)
plt.legend(title="Repository", bbox_to_anchor=(1.05, 1), loc="upper
left")
plt.xticks(ticks=range(0, len(weekly_pivot), 10),
labels=weekly_pivot.index[:10], rotation=45, fontsize=10)
plt.grid(axis="y", linestyle="--", alpha=0.7)

# Save or show the plot
plt.tight_layout()
plt.savefig("weekly_closed_issues_bar_chart.png")
plt.show()

```

```

C:\Users\soham\AppData\Local\Temp\ipykernel_18460\783876463.py:9:
DtypeWarning: Columns (5,6,7,8,9,10,12) have mixed types. Specify
dtype option on import or set low_memory=False.
    df_detailed = pd.read_csv("github_all_data.csv")
C:\Users\soham\AppData\Local\Temp\ipykernel_18460\783876463.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

```

See the caveats in the documentation:

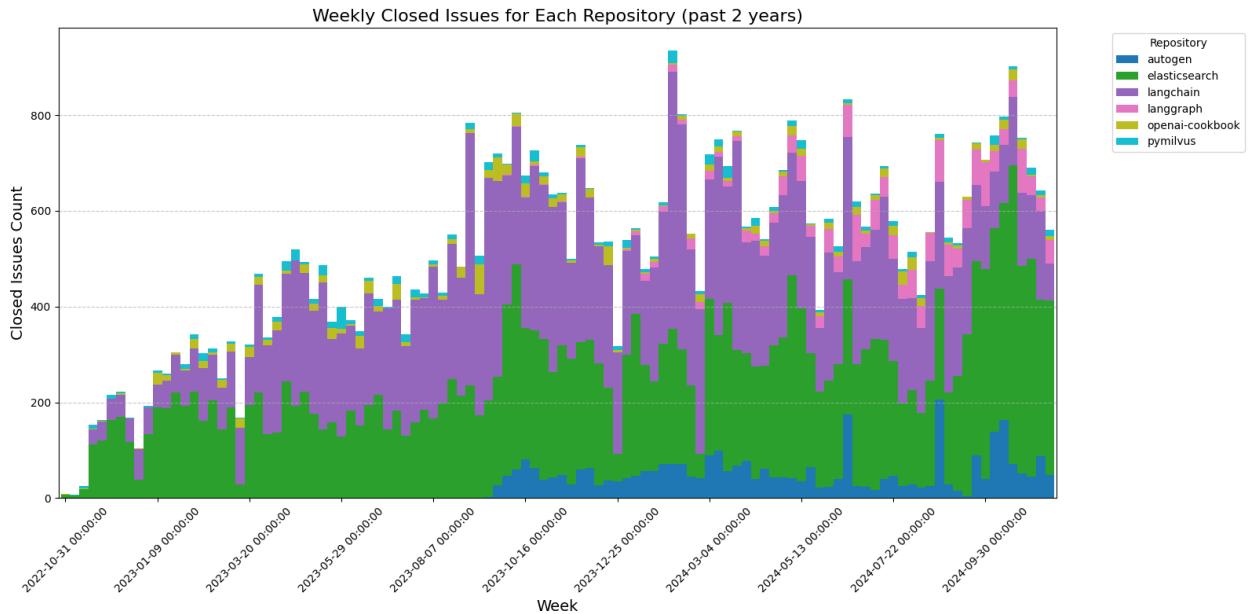
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```

df_issues["closed_at"] = pd.to_datetime(df_issues["closed_at"],
errors="coerce")
C:\Users\soham\AppData\Local\Temp\ipykernel_18460\783876463.py:17:
UserWarning: Converting to PeriodArray/Index representation will drop
timezone information.
    df_issues["week"] =
df_issues["closed_at"].dt.to_period("W").apply(lambda r: r.start_time)

<Figure size 1400x800 with 0 Axes>

```



```
#Add your code for requirement 7 in this cell
```

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter data for issues only
df_issues = df_detailed[df_detailed["type"] == "issue"]

# Convert to datetime and filter data to the specified range (if needed)
df_issues["created_at"] = pd.to_datetime(df_issues["created_at"],
                                         errors="coerce")
df_issues["closed_at"] = pd.to_datetime(df_issues["closed_at"],
                                         errors="coerce")

# Filter for issues created and closed between November 2022 to
# November 2024
df_issues = df_issues[(df_issues["created_at"] >= "2022-11-01") &
                      (df_issues["closed_at"] <= "2024-11-30")]

# Count created issues per repo
created_issues = df_issues.groupby("repository")[
    "created_at"].count().reset_index(name="created_issues")

# Count closed issues per repo (where closed_at is not null)
closed_issues =
df_issues[df_issues["closed_at"].notnull()].groupby("repository")[
    "closed_at"].count().reset_index(name="closed_issues")
```

```

# Merge created and closed issues into one dataframe
issues_summary = pd.merge(created_issues, closed_issues,
on="repository", how="left").fillna(0)

# Plot the stacked bar chart
plt.figure(figsize=(12, 6))
issues_summary.set_index("repository")[["created_issues",
"closed_issues"]].plot(kind="bar", stacked=True, figsize=(16, 8),
width=0.8, colormap="Set2")

# Add titles and labels
plt.title("Created and Closed Issues per Repository (2022-11 to 2024-11)", fontsize=16)
plt.xlabel("Repository", fontsize=14)
plt.ylabel("Issue Count", fontsize=14)
plt.xticks(rotation=45, ha="right", fontsize=10)
plt.legend(title="Issue Status", bbox_to_anchor=(1.05, 1), loc="upper left")
plt.grid(axis="y", linestyle="--", alpha=0.7)

# Save or show the plot
plt.tight_layout()
plt.savefig("created_vs_closed_issues_stack_bar_chart.png")
plt.show()

```

```

C:\Users\soham\AppData\Local\Temp\ipykernel_18460\2469595251.py:7:
DtypeWarning: Columns (5,6,7,8,9,10,12) have mixed types. Specify
dtype option on import or set low_memory=False.
    df_detailed = pd.read_csv("github_all_data.csv")
C:\Users\soham\AppData\Local\Temp\ipykernel_18460\2469595251.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

```

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```

    df_issues["created_at"] = pd.to_datetime(df_issues["created_at"],
errors="coerce")
C:\Users\soham\AppData\Local\Temp\ipykernel_18460\2469595251.py:14:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

```

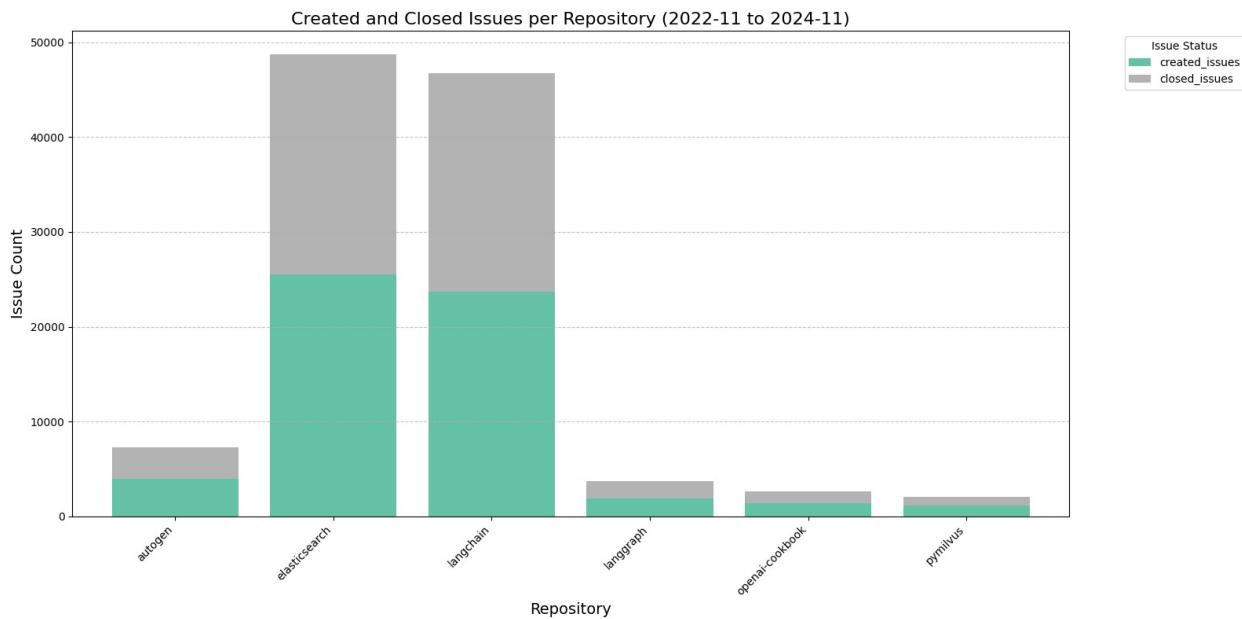
See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```

    df_issues["closed_at"] = pd.to_datetime(df_issues["closed_at"],
errors="coerce")

```

<Figure size 1200x600 with 0 Axes>



```
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime
from tensorflow.keras.preprocessing.sequence import TimeseriesGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter the data for 'issues' type
df_issues = df_detailed[df_detailed['type'] == 'issue']

# Convert 'created_at' to datetime
df_issues['created_at'] = pd.to_datetime(df_issues['created_at'])

# Extract the day of the week from 'created_at' (0=Monday, 6=Sunday)
df_issues['day_of_week'] = df_issues['created_at'].dt.dayofweek

# Group by repository and day of the week to count issues created
df_day_of_week = df_issues.groupby(['repository', 'day_of_week']).size().reset_index(name='issue_count')
```

```

# Get the average number of issues created for each day of the week
# across the entire period for each repo
df_avg_day_of_week = df_day_of_week.groupby(['repository',
'day_of_week'])['issue_count'].mean().reset_index()

# Pivot the data to have a row for each repository and columns for
# days of the week (0-6)
df_pivot = df_avg_day_of_week.pivot(index='repository',
columns='day_of_week', values='issue_count')

# Fill missing values with 0 (if any)
df_pivot = df_pivot.fillna(0)

# Check the data
#print(df_pivot.head())

# Prepare the time series data for LSTM
X = df_pivot.values # Each repo's issue count per day of the week (0-
6)
y = np.argmax(X, axis=1) # Forecast the day with the maximum issues
# (index of max value)

# Reshape the input for LSTM [samples, time steps, features]
X_reshaped = X.reshape((X.shape[0], X.shape[1], 1))

# Split data into training and testing sets (80% train, 20% test)
train_size = int(len(X_reshaped) * 0.8)
X_train, X_test = X_reshaped[:train_size], X_reshaped[train_size:]
y_train, y_test = y[:train_size], y[train_size:]

# Normalize the data (optional but recommended for LSTM)
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train.reshape(-1,
X_train.shape[1])).reshape(X_train.shape)
X_test = scaler.transform(X_test.reshape(-1,
X_test.shape[1])).reshape(X_test.shape)

# Build the LSTM model
model = Sequential()

# LSTM layer
model.add(LSTM(units=50, return_sequences=False,
input_shape=(X_train.shape[1], X_train.shape[2])))

# Output layer (predicting the day of the week with max issues)
model.add(Dense(units=7, activation='softmax')) # 7 possible days (0-
6)

# Compile the model

```

```

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Train the model
history = model.fit(X_train, y_train, epochs=20, batch_size=16,
                      validation_data=(X_test, y_test), verbose=1)

# Plot training history
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Model Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.savefig("forecasting/Tensorflow_LSTM/lstm_day_max_issues_created.png")
plt.show()

# Evaluate the model on the test set
test_loss, test_accuracy = model.evaluate(X_test, y_test)
print(f"Test Accuracy: {test_accuracy * 100:.2f}%")

# Make predictions for each repository in the dataset
y_pred = model.predict(X_reshaped)

# Convert predictions from one-hot encoding to day of the week (0-6)
predicted_days = np.argmax(y_pred, axis=1)

# Create a DataFrame to show predictions for all repositories
predictions_df_all_repos = pd.DataFrame({
    'repository': df_pivot.index, # All repositories
    'predicted_day_of_week': predicted_days,
})

# Map the predicted day number to the name of the day
day_mapping = {0: 'Monday', 1: 'Tuesday', 2: 'Wednesday', 3: 'Thursday', 4: 'Friday', 5: 'Saturday', 6: 'Sunday'}
predictions_df_all_repos['predicted_day_of_week'] =
    predictions_df_all_repos['predicted_day_of_week'].map(day_mapping)

# Display the predictions for all repositories
print(predictions_df_all_repos)

```

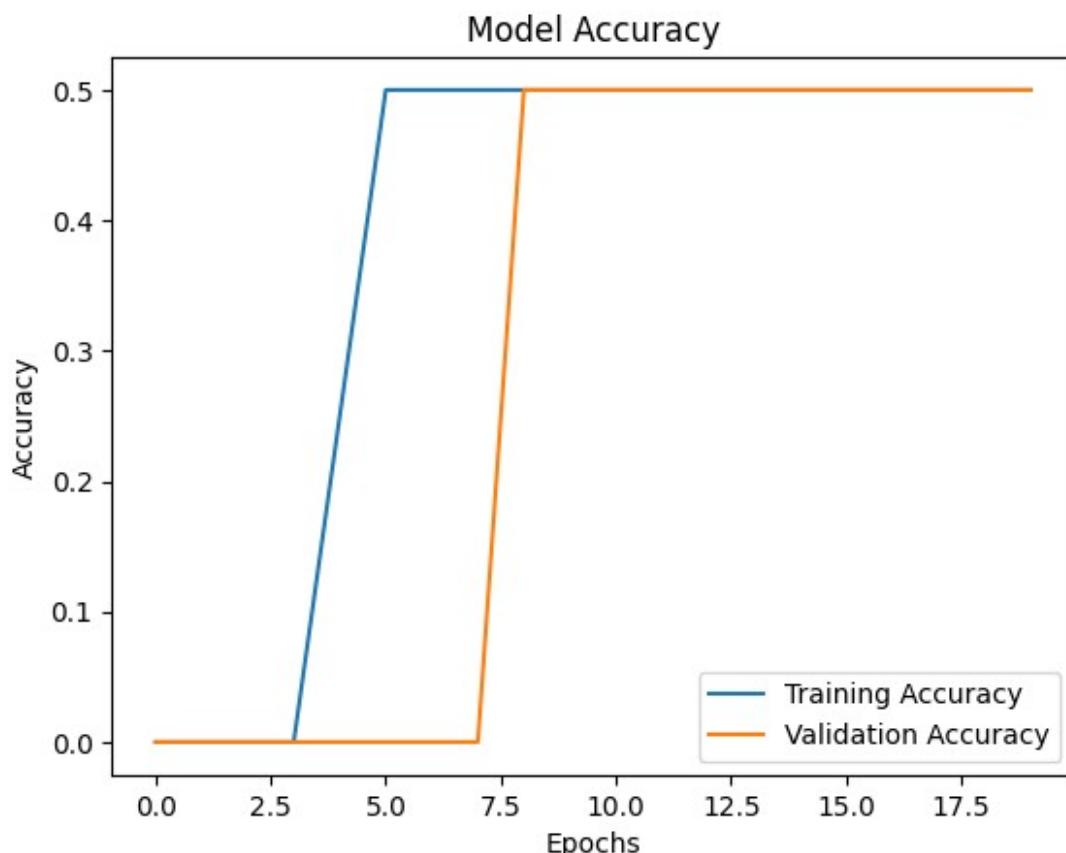
```

Epoch 1/20
1/1 ━━━━━━━━ 3s 3s/step - accuracy: 0.0000e+00 - loss: 1.9390 - val_accuracy: 0.0000e+00 - val_loss: 1.9414
Epoch 2/20
1/1 ━━━━━━━━ 0s 86ms/step - accuracy: 0.0000e+00 - loss: 1.9198 - val_accuracy: 0.0000e+00 - val_loss: 1.9364

```

```
Epoch 3/20
1/1 _____ 0s 70ms/step - accuracy: 0.0000e+00 - loss:
1.9006 - val_accuracy: 0.0000e+00 - val_loss: 1.9314
Epoch 4/20
1/1 _____ 0s 65ms/step - accuracy: 0.0000e+00 - loss:
1.8814 - val_accuracy: 0.0000e+00 - val_loss: 1.9264
Epoch 5/20
1/1 _____ 0s 68ms/step - accuracy: 0.2500 - loss:
1.8621 - val_accuracy: 0.0000e+00 - val_loss: 1.9214
Epoch 6/20
1/1 _____ 0s 78ms/step - accuracy: 0.5000 - loss:
1.8427 - val_accuracy: 0.0000e+00 - val_loss: 1.9164
Epoch 7/20
1/1 _____ 0s 67ms/step - accuracy: 0.5000 - loss:
1.8230 - val_accuracy: 0.0000e+00 - val_loss: 1.9113
Epoch 8/20
1/1 _____ 0s 70ms/step - accuracy: 0.5000 - loss:
1.8031 - val_accuracy: 0.0000e+00 - val_loss: 1.9061
Epoch 9/20
1/1 _____ 0s 77ms/step - accuracy: 0.5000 - loss:
1.7828 - val_accuracy: 0.5000 - val_loss: 1.9007
Epoch 10/20
1/1 _____ 0s 77ms/step - accuracy: 0.5000 - loss:
1.7621 - val_accuracy: 0.5000 - val_loss: 1.8953
Epoch 11/20
1/1 _____ 0s 82ms/step - accuracy: 0.5000 - loss:
1.7408 - val_accuracy: 0.5000 - val_loss: 1.8897
Epoch 12/20
1/1 _____ 0s 79ms/step - accuracy: 0.5000 - loss:
1.7189 - val_accuracy: 0.5000 - val_loss: 1.8839
Epoch 13/20
1/1 _____ 0s 81ms/step - accuracy: 0.5000 - loss:
1.6963 - val_accuracy: 0.5000 - val_loss: 1.8778
Epoch 14/20
1/1 _____ 0s 83ms/step - accuracy: 0.5000 - loss:
1.6729 - val_accuracy: 0.5000 - val_loss: 1.8716
Epoch 15/20
1/1 _____ 0s 76ms/step - accuracy: 0.5000 - loss:
1.6487 - val_accuracy: 0.5000 - val_loss: 1.8651
Epoch 16/20
1/1 _____ 0s 79ms/step - accuracy: 0.5000 - loss:
1.6235 - val_accuracy: 0.5000 - val_loss: 1.8582
Epoch 17/20
1/1 _____ 0s 70ms/step - accuracy: 0.5000 - loss:
1.5972 - val_accuracy: 0.5000 - val_loss: 1.8511
Epoch 18/20
1/1 _____ 0s 66ms/step - accuracy: 0.5000 - loss:
1.5697 - val_accuracy: 0.5000 - val_loss: 1.8435
Epoch 19/20
```

```
1/1 ━━━━━━━━━━ 0s 65ms/step - accuracy: 0.5000 - loss:  
1.5411 - val_accuracy: 0.5000 - val_loss: 1.8355  
Epoch 20/20  
1/1 ━━━━━━━━━━ 0s 64ms/step - accuracy: 0.5000 - loss:  
1.5111 - val_accuracy: 0.5000 - val_loss: 1.8271
```



```
1/1 ━━━━━━━━━━ 0s 31ms/step - accuracy: 0.5000 - loss:  
1.8271  
Test Accuracy: 50.00%  
1/1 ━━━━━━━━━━ 0s 211ms/step  
repository predicted_day_of_week  
0 autogen Tuesday  
1 elasticsearch Tuesday  
2 langchain Tuesday  
3 langgraph Tuesday  
4 openai-cookbook Tuesday  
5 pymilvus Tuesday
```

#Add your code for requirement 8.2 in this cell

```
import pandas as pd  
import numpy as np
```

```

import matplotlib.pyplot as plt
from datetime import datetime
from tensorflow.keras.preprocessing.sequence import TimeseriesGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter the data for 'issue' type and 'closed' status
df_closed_issues = df_detailed[(df_detailed['type'] == 'issue') & (df_detailed['state'] == 'closed')]

# Convert 'created_at' to datetime
df_closed_issues['created_at'] =
pd.to_datetime(df_closed_issues['created_at'])

# Extract the day of the week from 'created_at' (0=Monday, 6=Sunday)
df_closed_issues['day_of_week'] =
df_closed_issues['created_at'].dt.dayofweek

# Group by repository and day of the week to count closed issues
df_day_of_week_closed = df_closed_issues.groupby(['repository',
'day_of_week']).size().reset_index(name='closed_issue_count')

# Get the average number of closed issues for each day of the week
# across the entire period for each repo
df_avg_day_of_week_closed =
df_day_of_week_closed.groupby(['repository', 'day_of_week'])
['closed_issue_count'].mean().reset_index()

# Pivot the data to have a row for each repository and columns for
# days of the week (0-6)
df_pivot_closed = df_avg_day_of_week_closed.pivot(index='repository',
columns='day_of_week', values='closed_issue_count')

# Fill missing values with 0 (if any)
df_pivot_closed = df_pivot_closed.fillna(0)

# Check the data
#print(df_pivot_closed.head())

# Prepare the time series data for LSTM
X_closed = df_pivot_closed.values # Each repo's closed issue count
# per day of the week (0-6)
y_closed = np.argmax(X_closed, axis=1) # Forecast the day with the
# maximum closed issues (index of max value)

```

```

# Reshape the input for LSTM [samples, time steps, features]
X_reshaped_closed = X_closed.reshape((X_closed.shape[0],
X_closed.shape[1], 1))

# Split data into training and testing sets (80% train, 20% test)
train_size_closed = int(len(X_reshaped_closed) * 0.8)
X_train_closed, X_test_closed = X_reshaped_closed[:train_size_closed],
X_reshaped_closed[train_size_closed:]
y_train_closed, y_test_closed = y_closed[:train_size_closed],
y_closed[train_size_closed:]

# Normalize the data (optional but recommended for LSTM)
scaler_closed = MinMaxScaler()
X_train_closed = scaler_closed.fit_transform(X_train_closed.reshape(-1,
X_train_closed.shape[1])).reshape(X_train_closed.shape)
X_test_closed = scaler_closed.transform(X_test_closed.reshape(-1,
X_test_closed.shape[1])).reshape(X_test_closed.shape)

# Build the LSTM model
model_closed = Sequential()

# LSTM layer
model_closed.add(LSTM(units=50, return_sequences=False,
input_shape=(X_train_closed.shape[1], X_train_closed.shape[2])))

# Output layer (predicting the day of the week with max closed issues)
model_closed.add(Dense(units=7, activation='softmax')) # 7 possible
days (0-6)

# Compile the model
model_closed.compile(optimizer='adam',
loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Train the model
history_closed = model_closed.fit(X_train_closed, y_train_closed,
epochs=50, batch_size=16, validation_data=(X_test_closed,
y_test_closed), verbose=1)

# Plot training history
plt.plot(history_closed.history['accuracy'], label='Training
Accuracy')
plt.plot(history_closed.history['val_accuracy'], label='Validation
Accuracy')
plt.title('Model Accuracy for Closed Issues')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.savefig("forecasting/Tensorflow_LSTM/lstm_day_max_issues_closed.pn
g")
plt.show()

```

```

# Evaluate the model on the test set
test_loss_closed, test_accuracy_closed =
model_closed.evaluate(X_test_closed, y_test_closed)
print(f"Test Accuracy for Closed Issues: {test_accuracy_closed * 100:.2f}%")

# Make predictions for each repository in the dataset
y_pred_closed = model_closed.predict(X_reshaped_closed)

# Convert predictions from one-hot encoding to day of the week (0-6)
predicted_days_closed = np.argmax(y_pred_closed, axis=1)

# Create a DataFrame to show predictions for all repositories
predictions_df_all_repos_closed = pd.DataFrame({
    'repository': df_pivot_closed.index, # All repositories
    'predicted_day_of_week': predicted_days_closed,
})

# Map the predicted day number to the name of the day
predictions_df_all_repos_closed['predicted_day_of_week'] =
predictions_df_all_repos_closed['predicted_day_of_week'].map(day_mapping)

# Display the predictions for all repositories
print(predictions_df_all_repos_closed)

```

```

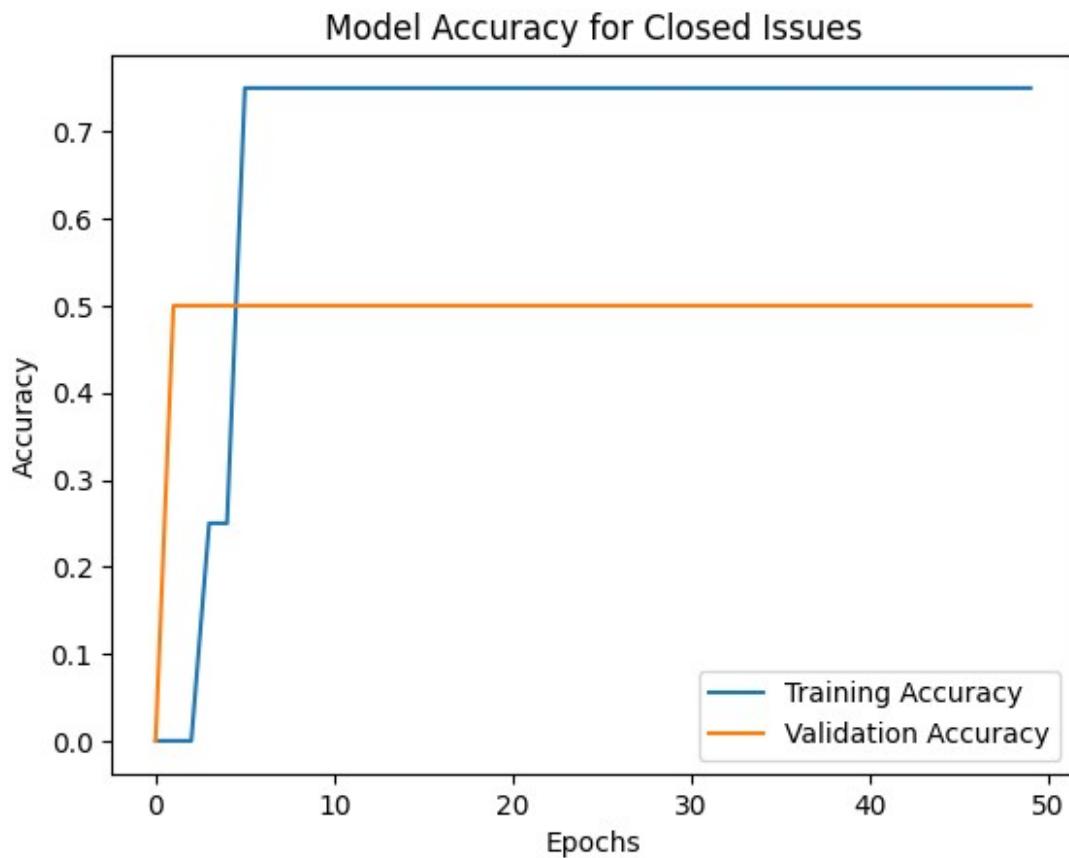
Epoch 1/50
1/1 ██████████ 3s 3s/step - accuracy: 0.0000e+00 - loss:
1.9454 - val_accuracy: 0.0000e+00 - val_loss: 1.9428
Epoch 2/50
1/1 █████ 0s 76ms/step - accuracy: 0.0000e+00 - loss:
1.9265 - val_accuracy: 0.5000 - val_loss: 1.9385
Epoch 3/50
1/1 █████ 0s 68ms/step - accuracy: 0.0000e+00 - loss:
1.9076 - val_accuracy: 0.5000 - val_loss: 1.9341
Epoch 4/50
1/1 █████ 0s 86ms/step - accuracy: 0.2500 - loss:
1.8886 - val_accuracy: 0.5000 - val_loss: 1.9295
Epoch 5/50
1/1 █████ 0s 78ms/step - accuracy: 0.2500 - loss:
1.8694 - val_accuracy: 0.5000 - val_loss: 1.9248
Epoch 6/50
1/1 █████ 0s 83ms/step - accuracy: 0.7500 - loss:
1.8501 - val_accuracy: 0.5000 - val_loss: 1.9198
Epoch 7/50
1/1 █████ 0s 86ms/step - accuracy: 0.7500 - loss:
1.8304 - val_accuracy: 0.5000 - val_loss: 1.9147

```

```
Epoch 8/50
1/1 _____ 0s 85ms/step - accuracy: 0.7500 - loss:
1.8103 - val_accuracy: 0.5000 - val_loss: 1.9094
Epoch 9/50
1/1 _____ 0s 96ms/step - accuracy: 0.7500 - loss:
1.7897 - val_accuracy: 0.5000 - val_loss: 1.9039
Epoch 10/50
1/1 _____ 0s 97ms/step - accuracy: 0.7500 - loss:
1.7685 - val_accuracy: 0.5000 - val_loss: 1.8981
Epoch 11/50
1/1 _____ 0s 74ms/step - accuracy: 0.7500 - loss:
1.7466 - val_accuracy: 0.5000 - val_loss: 1.8921
Epoch 12/50
1/1 _____ 0s 68ms/step - accuracy: 0.7500 - loss:
1.7239 - val_accuracy: 0.5000 - val_loss: 1.8859
Epoch 13/50
1/1 _____ 0s 74ms/step - accuracy: 0.7500 - loss:
1.7005 - val_accuracy: 0.5000 - val_loss: 1.8794
Epoch 14/50
1/1 _____ 0s 80ms/step - accuracy: 0.7500 - loss:
1.6761 - val_accuracy: 0.5000 - val_loss: 1.8726
Epoch 15/50
1/1 _____ 0s 83ms/step - accuracy: 0.7500 - loss:
1.6506 - val_accuracy: 0.5000 - val_loss: 1.8654
Epoch 16/50
1/1 _____ 0s 72ms/step - accuracy: 0.7500 - loss:
1.6241 - val_accuracy: 0.5000 - val_loss: 1.8579
Epoch 17/50
1/1 _____ 0s 91ms/step - accuracy: 0.7500 - loss:
1.5963 - val_accuracy: 0.5000 - val_loss: 1.8499
Epoch 18/50
1/1 _____ 0s 87ms/step - accuracy: 0.7500 - loss:
1.5673 - val_accuracy: 0.5000 - val_loss: 1.8415
Epoch 19/50
1/1 _____ 0s 86ms/step - accuracy: 0.7500 - loss:
1.5368 - val_accuracy: 0.5000 - val_loss: 1.8326
Epoch 20/50
1/1 _____ 0s 85ms/step - accuracy: 0.7500 - loss:
1.5049 - val_accuracy: 0.5000 - val_loss: 1.8232
Epoch 21/50
1/1 _____ 0s 101ms/step - accuracy: 0.7500 - loss:
1.4713 - val_accuracy: 0.5000 - val_loss: 1.8131
Epoch 22/50
1/1 _____ 0s 160ms/step - accuracy: 0.7500 - loss:
1.4361 - val_accuracy: 0.5000 - val_loss: 1.8023
Epoch 23/50
1/1 _____ 0s 73ms/step - accuracy: 0.7500 - loss:
1.3992 - val_accuracy: 0.5000 - val_loss: 1.7908
Epoch 24/50
```

```
1/1 ━━━━━━━━ 0s 56ms/step - accuracy: 0.7500 - loss:  
1.3605 - val_accuracy: 0.5000 - val_loss: 1.7784  
Epoch 25/50  
1/1 ━━━━━━━━ 0s 62ms/step - accuracy: 0.7500 - loss:  
1.3201 - val_accuracy: 0.5000 - val_loss: 1.7651  
Epoch 26/50  
1/1 ━━━━━━━━ 0s 64ms/step - accuracy: 0.7500 - loss:  
1.2781 - val_accuracy: 0.5000 - val_loss: 1.7508  
Epoch 27/50  
1/1 ━━━━━━━━ 0s 78ms/step - accuracy: 0.7500 - loss:  
1.2346 - val_accuracy: 0.5000 - val_loss: 1.7354  
Epoch 28/50  
1/1 ━━━━━━━━ 0s 57ms/step - accuracy: 0.7500 - loss:  
1.1899 - val_accuracy: 0.5000 - val_loss: 1.7188  
Epoch 29/50  
1/1 ━━━━━━━━ 0s 58ms/step - accuracy: 0.7500 - loss:  
1.1443 - val_accuracy: 0.5000 - val_loss: 1.7009  
Epoch 30/50  
1/1 ━━━━━━━━ 0s 86ms/step - accuracy: 0.7500 - loss:  
1.0983 - val_accuracy: 0.5000 - val_loss: 1.6817  
Epoch 31/50  
1/1 ━━━━━━━━ 0s 82ms/step - accuracy: 0.7500 - loss:  
1.0523 - val_accuracy: 0.5000 - val_loss: 1.6610  
Epoch 32/50  
1/1 ━━━━━━━━ 0s 111ms/step - accuracy: 0.7500 - loss:  
1.0069 - val_accuracy: 0.5000 - val_loss: 1.6388  
Epoch 33/50  
1/1 ━━━━━━━━ 0s 75ms/step - accuracy: 0.7500 - loss:  
0.9628 - val_accuracy: 0.5000 - val_loss: 1.6152  
Epoch 34/50  
1/1 ━━━━━━━━ 0s 74ms/step - accuracy: 0.7500 - loss:  
0.9204 - val_accuracy: 0.5000 - val_loss: 1.5900  
Epoch 35/50  
1/1 ━━━━━━━━ 0s 85ms/step - accuracy: 0.7500 - loss:  
0.8802 - val_accuracy: 0.5000 - val_loss: 1.5633  
Epoch 36/50  
1/1 ━━━━━━━━ 0s 66ms/step - accuracy: 0.7500 - loss:  
0.8425 - val_accuracy: 0.5000 - val_loss: 1.5352  
Epoch 37/50  
1/1 ━━━━━━━━ 0s 91ms/step - accuracy: 0.7500 - loss:  
0.8075 - val_accuracy: 0.5000 - val_loss: 1.5059  
Epoch 38/50  
1/1 ━━━━━━━━ 0s 62ms/step - accuracy: 0.7500 - loss:  
0.7752 - val_accuracy: 0.5000 - val_loss: 1.4757  
Epoch 39/50  
1/1 ━━━━━━━━ 0s 55ms/step - accuracy: 0.7500 - loss:  
0.7457 - val_accuracy: 0.5000 - val_loss: 1.4448  
Epoch 40/50  
1/1 ━━━━━━━━ 0s 68ms/step - accuracy: 0.7500 - loss:
```

```
0.7188 - val_accuracy: 0.5000 - val_loss: 1.4139
Epoch 41/50
1/1 ━━━━━━━━ 0s 76ms/step - accuracy: 0.7500 - loss:
0.6946 - val_accuracy: 0.5000 - val_loss: 1.3833
Epoch 42/50
1/1 ━━━━━━━━ 0s 87ms/step - accuracy: 0.7500 - loss:
0.6730 - val_accuracy: 0.5000 - val_loss: 1.3537
Epoch 43/50
1/1 ━━━━━━━━ 0s 162ms/step - accuracy: 0.7500 - loss:
0.6540 - val_accuracy: 0.5000 - val_loss: 1.3258
Epoch 44/50
1/1 ━━━━━━━━ 0s 96ms/step - accuracy: 0.7500 - loss:
0.6376 - val_accuracy: 0.5000 - val_loss: 1.2999
Epoch 45/50
1/1 ━━━━━━━━ 0s 79ms/step - accuracy: 0.7500 - loss:
0.6239 - val_accuracy: 0.5000 - val_loss: 1.2765
Epoch 46/50
1/1 ━━━━━━━━ 0s 77ms/step - accuracy: 0.7500 - loss:
0.6126 - val_accuracy: 0.5000 - val_loss: 1.2553
Epoch 47/50
1/1 ━━━━━━━━ 0s 76ms/step - accuracy: 0.7500 - loss:
0.6033 - val_accuracy: 0.5000 - val_loss: 1.2362
Epoch 48/50
1/1 ━━━━━━━━ 0s 57ms/step - accuracy: 0.7500 - loss:
0.5956 - val_accuracy: 0.5000 - val_loss: 1.2184
Epoch 49/50
1/1 ━━━━━━━━ 0s 59ms/step - accuracy: 0.7500 - loss:
0.5888 - val_accuracy: 0.5000 - val_loss: 1.2012
Epoch 50/50
1/1 ━━━━━━━━ 0s 61ms/step - accuracy: 0.7500 - loss:
0.5823 - val_accuracy: 0.5000 - val_loss: 1.1837
```



```

1/1 ━━━━━━━━━━ 0s 56ms/step - accuracy: 0.5000 - loss:
1.1837
Test Accuracy for Closed Issues: 50.00%
1/1 ━━━━━━━━ 0s 235ms/step
      repository predicted_day_of_week
0      autogen          Tuesday
1  elasticsearch          Tuesday
2    langchain          Tuesday
3    langgraph          Tuesday
4 openai-cookbook          Tuesday
5     pymilvus          Tuesday

import pandas as pd
import numpy as np
from datetime import datetime
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter the data for 'issue' type and 'closed' status

```

```

df_closed_issues = df_detailed[(df_detailed['type'] == 'issue') &
                               (df_detailed['state'] == 'closed')]

# Convert 'created_at' to datetime
df_closed_issues['created_at'] =
    pd.to_datetime(df_closed_issues['created_at'])

# Extract the month of the year (1=January, 12=December)
df_closed_issues['month_of_year'] =
    df_closed_issues['created_at'].dt.month

# Group by repository and month of the year to count closed issues
df_month_of_year_closed = df_closed_issues.groupby(['repository',
                                                     'month_of_year']).size().reset_index(name='closed_issue_count')

# Get the average number of closed issues for each month of the year
# across the entire period for each repo
df_avg_month_of_year_closed =
    df_month_of_year_closed.groupby(['repository', 'month_of_year'])[
        'closed_issue_count'].mean().reset_index()

# Pivot the data to have a row for each repository and columns for
# months of the year (1-12)
df_pivot_closed =
    df_avg_month_of_year_closed.pivot(index='repository',
                                       columns='month_of_year', values='closed_issue_count')

# Fill missing values with 0 (if any)
df_pivot_closed = df_pivot_closed.fillna(0)

# Check the data
#print(df_pivot_closed.head())

# Prepare the time series data for LSTM
X_closed_month = df_pivot_closed.values # Each repo's closed issue
                                         # count per month (1-12)
y_closed_month = np.argmax(X_closed_month, axis=1) # Forecast the
                                         # month with the maximum closed issues (index of max value)

# Reshape the input for LSTM [samples, time steps, features]
X_reshaped_month = X_closed_month.reshape((X_closed_month.shape[0],
                                             X_closed_month.shape[1], 1))

# Split data into training and testing sets (80% train, 20% test)
train_size_month = int(len(X_reshaped_month) * 0.8)
X_train_month, X_test_month = X_reshaped_month[:train_size_month],
X_reshaped_month[train_size_month:]
y_train_month, y_test_month = y_closed_month[:train_size_month],
y_closed_month[train_size_month:]

```

```

# Normalize the data (optional but recommended for LSTM)
scaler_month = MinMaxScaler()
X_train_month = scaler_month.fit_transform(X_train_month.reshape(-1,
X_train_month.shape[1])).reshape(X_train_month.shape)
X_test_month = scaler_month.transform(X_test_month.reshape(-1,
X_test_month.shape[1])).reshape(X_test_month.shape)

# Build the LSTM model
model_month = Sequential()

# LSTM layer
model_month.add(LSTM(units=50, return_sequences=False,
input_shape=(X_train_month.shape[1], X_train_month.shape[2])))

# Output layer (predicting the month of the year with max closed
issues)
model_month.add(Dense(units=12, activation='softmax')) # 12 possible
months (1-12)

# Compile the model
model_month.compile(optimizer='adam',
loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Train the model
history_month = model_month.fit(X_train_month, y_train_month,
epochs=50, batch_size=16, validation_data=(X_test_month,
y_test_month), verbose=1)

# Evaluate the model on the test set
test_loss_month, test_accuracy_month =
model_month.evaluate(X_test_month, y_test_month)
#print(f"Test Accuracy for Closed Issues by Month:
{test_accuracy_month * 100:.2f}%")

# Make predictions for each repository in the dataset
y_pred_month = model_month.predict(X_reshaped_month)

# Convert predictions from one-hot encoding to month of the year (1-
12)
predicted_months = np.argmax(y_pred_month, axis=1) + 1 # +1 to map to
months (1-12)

# Create a DataFrame to show predictions for all repositories
predictions_df_all_repos_month = pd.DataFrame({
    'repository': df_pivot_closed.index, # All repositories
    'predicted_month_of_year': predicted_months,
})

# Map the predicted month number to the name of the month
month_mapping = {1: 'January', 2: 'February', 3: 'March', 4: 'April',
5: 'May', 6: 'June', 7: 'July', 8: 'August', 9: 'September', 10: 'October',
11: 'November', 12: 'December'}

```

```
5: 'May', 6: 'June', 7: 'July', 8: 'August',
         9: 'September', 10: 'October', 11: 'November', 12:
'December'}
predictions_df_all_repos_month['predicted_month_of_year'] =
predictions_df_all_repos_month['predicted_month_of_year'].map(month_mapping)

# Display the predictions for all repositories
print(predictions_df_all_repos_month)

Epoch 1/50
1/1 ━━━━━━━━━━ 2s 2s/step - accuracy: 0.0000e+00 - loss:
2.4563 - val_accuracy: 0.0000e+00 - val_loss: 2.4864
Epoch 2/50
1/1 ━━━━━━━━━━ 0s 66ms/step - accuracy: 0.2500 - loss:
2.4395 - val_accuracy: 0.0000e+00 - val_loss: 2.4879
Epoch 3/50
1/1 ━━━━━━━━━━ 0s 68ms/step - accuracy: 0.5000 - loss:
2.4228 - val_accuracy: 0.0000e+00 - val_loss: 2.4894
Epoch 4/50
1/1 ━━━━━━━━━━ 0s 66ms/step - accuracy: 0.5000 - loss:
2.4061 - val_accuracy: 0.0000e+00 - val_loss: 2.4908
Epoch 5/50
1/1 ━━━━━━━━━━ 0s 68ms/step - accuracy: 0.5000 - loss:
2.3892 - val_accuracy: 0.0000e+00 - val_loss: 2.4922
Epoch 6/50
1/1 ━━━━━━━━━━ 0s 70ms/step - accuracy: 0.5000 - loss:
2.3722 - val_accuracy: 0.0000e+00 - val_loss: 2.4937
Epoch 7/50
1/1 ━━━━━━━━━━ 0s 73ms/step - accuracy: 0.5000 - loss:
2.3549 - val_accuracy: 0.0000e+00 - val_loss: 2.4952
Epoch 8/50
1/1 ━━━━━━━━━━ 0s 64ms/step - accuracy: 0.5000 - loss:
2.3371 - val_accuracy: 0.0000e+00 - val_loss: 2.4969
Epoch 9/50
1/1 ━━━━━━━━━━ 0s 74ms/step - accuracy: 0.5000 - loss:
2.3186 - val_accuracy: 0.0000e+00 - val_loss: 2.4987
Epoch 10/50
1/1 ━━━━━━━━━━ 0s 72ms/step - accuracy: 0.5000 - loss:
2.2993 - val_accuracy: 0.0000e+00 - val_loss: 2.5007
Epoch 11/50
1/1 ━━━━━━━━━━ 0s 68ms/step - accuracy: 0.5000 - loss:
2.2789 - val_accuracy: 0.0000e+00 - val_loss: 2.5028
Epoch 12/50
1/1 ━━━━━━━━━━ 0s 68ms/step - accuracy: 0.5000 - loss:
2.2571 - val_accuracy: 0.0000e+00 - val_loss: 2.5050
Epoch 13/50
1/1 ━━━━━━━━━━ 0s 65ms/step - accuracy: 0.5000 - loss:
2.2338 - val_accuracy: 0.0000e+00 - val_loss: 2.5074
Epoch 14/50
```

```
1/1 ━━━━━━━━ 0s 66ms/step - accuracy: 0.5000 - loss:  
2.2085 - val_accuracy: 0.0000e+00 - val_loss: 2.5100  
Epoch 15/50  
1/1 ━━━━━━━━ 0s 70ms/step - accuracy: 0.5000 - loss:  
2.1809 - val_accuracy: 0.0000e+00 - val_loss: 2.5127  
Epoch 16/50  
1/1 ━━━━━━━━ 0s 89ms/step - accuracy: 0.5000 - loss:  
2.1508 - val_accuracy: 0.0000e+00 - val_loss: 2.5157  
Epoch 17/50  
1/1 ━━━━━━━━ 0s 71ms/step - accuracy: 0.5000 - loss:  
2.1177 - val_accuracy: 0.0000e+00 - val_loss: 2.5189  
Epoch 18/50  
1/1 ━━━━━━━━ 0s 72ms/step - accuracy: 0.5000 - loss:  
2.0813 - val_accuracy: 0.0000e+00 - val_loss: 2.5225  
Epoch 19/50  
1/1 ━━━━━━━━ 0s 64ms/step - accuracy: 0.5000 - loss:  
2.0413 - val_accuracy: 0.0000e+00 - val_loss: 2.5264  
Epoch 20/50  
1/1 ━━━━━━━━ 0s 70ms/step - accuracy: 0.5000 - loss:  
1.9975 - val_accuracy: 0.0000e+00 - val_loss: 2.5309  
Epoch 21/50  
1/1 ━━━━━━━━ 0s 63ms/step - accuracy: 0.5000 - loss:  
1.9498 - val_accuracy: 0.0000e+00 - val_loss: 2.5360  
Epoch 22/50  
1/1 ━━━━━━━━ 0s 72ms/step - accuracy: 0.5000 - loss:  
1.8983 - val_accuracy: 0.0000e+00 - val_loss: 2.5420  
Epoch 23/50  
1/1 ━━━━━━━━ 0s 75ms/step - accuracy: 0.5000 - loss:  
1.8432 - val_accuracy: 0.0000e+00 - val_loss: 2.5491  
Epoch 24/50  
1/1 ━━━━━━━━ 0s 116ms/step - accuracy: 0.5000 - loss:  
1.7853 - val_accuracy: 0.0000e+00 - val_loss: 2.5577  
Epoch 25/50  
1/1 ━━━━━━━━ 0s 65ms/step - accuracy: 0.5000 - loss:  
1.7255 - val_accuracy: 0.0000e+00 - val_loss: 2.5684  
Epoch 26/50  
1/1 ━━━━━━━━ 0s 75ms/step - accuracy: 0.5000 - loss:  
1.6649 - val_accuracy: 0.0000e+00 - val_loss: 2.5820  
Epoch 27/50  
1/1 ━━━━━━━━ 0s 93ms/step - accuracy: 0.5000 - loss:  
1.6044 - val_accuracy: 0.0000e+00 - val_loss: 2.5997  
Epoch 28/50  
1/1 ━━━━━━━━ 0s 84ms/step - accuracy: 0.5000 - loss:  
1.5448 - val_accuracy: 0.0000e+00 - val_loss: 2.6230  
Epoch 29/50  
1/1 ━━━━━━━━ 0s 86ms/step - accuracy: 0.5000 - loss:  
1.4863 - val_accuracy: 0.0000e+00 - val_loss: 2.6546  
Epoch 30/50  
1/1 ━━━━━━━━ 0s 188ms/step - accuracy: 0.5000 - loss:
```

```
1.4288 - val_accuracy: 0.0000e+00 - val_loss: 2.6978
Epoch 31/50
1/1 _____ 0s 88ms/step - accuracy: 0.5000 - loss:
1.3724 - val_accuracy: 0.0000e+00 - val_loss: 2.7575
Epoch 32/50
1/1 _____ 0s 72ms/step - accuracy: 0.5000 - loss:
1.3179 - val_accuracy: 0.0000e+00 - val_loss: 2.8400
Epoch 33/50
1/1 _____ 0s 70ms/step - accuracy: 0.5000 - loss:
1.2676 - val_accuracy: 0.0000e+00 - val_loss: 2.9526
Epoch 34/50
1/1 _____ 0s 74ms/step - accuracy: 0.5000 - loss:
1.2249 - val_accuracy: 0.0000e+00 - val_loss: 3.1006
Epoch 35/50
1/1 _____ 0s 76ms/step - accuracy: 0.5000 - loss:
1.1936 - val_accuracy: 0.0000e+00 - val_loss: 3.2826
Epoch 36/50
1/1 _____ 0s 66ms/step - accuracy: 0.5000 - loss:
1.1754 - val_accuracy: 0.0000e+00 - val_loss: 3.4865
Epoch 37/50
1/1 _____ 0s 71ms/step - accuracy: 0.5000 - loss:
1.1682 - val_accuracy: 0.0000e+00 - val_loss: 3.6917
Epoch 38/50
1/1 _____ 0s 67ms/step - accuracy: 0.5000 - loss:
1.1666 - val_accuracy: 0.0000e+00 - val_loss: 3.8782
Epoch 39/50
1/1 _____ 0s 112ms/step - accuracy: 0.5000 - loss:
1.1648 - val_accuracy: 0.0000e+00 - val_loss: 4.0325
Epoch 40/50
1/1 _____ 0s 107ms/step - accuracy: 0.5000 - loss:
1.1586 - val_accuracy: 0.0000e+00 - val_loss: 4.1487
Epoch 41/50
1/1 _____ 0s 75ms/step - accuracy: 0.5000 - loss:
1.1463 - val_accuracy: 0.0000e+00 - val_loss: 4.2259
Epoch 42/50
1/1 _____ 0s 83ms/step - accuracy: 0.5000 - loss:
1.1279 - val_accuracy: 0.0000e+00 - val_loss: 4.2667
Epoch 43/50
1/1 _____ 0s 78ms/step - accuracy: 0.5000 - loss:
1.1044 - val_accuracy: 0.0000e+00 - val_loss: 4.2755
Epoch 44/50
1/1 _____ 0s 171ms/step - accuracy: 0.5000 - loss:
1.0775 - val_accuracy: 0.0000e+00 - val_loss: 4.2579
Epoch 45/50
1/1 _____ 0s 66ms/step - accuracy: 0.5000 - loss:
1.0492 - val_accuracy: 0.0000e+00 - val_loss: 4.2206
Epoch 46/50
1/1 _____ 0s 66ms/step - accuracy: 0.5000 - loss:
1.0217 - val_accuracy: 0.0000e+00 - val_loss: 4.1709
```

```

Epoch 47/50
1/1 ━━━━━━━━ 0s 50ms/step - accuracy: 0.5000 - loss: 0.9966 - val_accuracy: 0.0000e+00 - val_loss: 4.1162
Epoch 48/50
1/1 ━━━━━━━━ 0s 65ms/step - accuracy: 0.5000 - loss: 0.9755 - val_accuracy: 0.0000e+00 - val_loss: 4.0636
Epoch 49/50
1/1 ━━━━━━━━ 0s 64ms/step - accuracy: 0.5000 - loss: 0.9588 - val_accuracy: 0.0000e+00 - val_loss: 4.0190
Epoch 50/50
1/1 ━━━━━━━━ 0s 65ms/step - accuracy: 0.5000 - loss: 0.9464 - val_accuracy: 0.0000e+00 - val_loss: 3.9874
1/1 ━━━━━━━━ 0s 22ms/step - accuracy: 0.0000e+00 - loss: 3.9874
Test Accuracy for Closed Issues by Month: 0.00%
1/1 ━━━━━━ 0s 160ms/step
      repository predicted_month_of_year
0      autogen          October
1  elasticsearch        April
2    langchain          April
3   langgraph          October
4 openai-cookbook      October
5     pymilvus          October

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter the data for 'issue' type and 'created' status
df_created_issues = df_detailed[df_detailed['type'] == 'issue']

# Convert 'created_at' to datetime
df_created_issues['created_at'] =
pd.to_datetime(df_created_issues['created_at'])

# Filter data between November 2022 and November 2024
start_date = '2022-11-01'
end_date = '2024-11-30'
df_created_issues = df_created_issues[(df_created_issues['created_at'] >= start_date) & (df_created_issues['created_at'] <= end_date)]

# Extract the date part (ignoring time) and group by date to count created issues

```

```

df_created_issues['date'] = df_created_issues['created_at'].dt.date
df_created_daily =
df_created_issues.groupby('date').size().reset_index(name='created_issue_count')

# Plot the original created issues over time (for visualization)
plt.figure(figsize=(10, 6))
plt.plot(df_created_daily['date'],
df_created_daily['created_issue_count'], label='Created Issues')
plt.xlabel('Date')
plt.ylabel('Number of Created Issues')
plt.title('Created Issues Over Time (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.show()

# Prepare the time series data for LSTM
created_issues_values =
df_created_daily['created_issue_count'].values.reshape(-1, 1)

# Normalize the data (MinMaxScaler)
scaler_created_issues = MinMaxScaler(feature_range=(0, 1))
created_issues_scaled =
scaler_created_issues.fit_transform(created_issues_values)

# Split data into training and testing sets (80% train, 20% test)
train_size = int(len(created_issues_scaled) * 0.8)
train_data, test_data = created_issues_scaled[:train_size],
created_issues_scaled[train_size:]

# Prepare the data for LSTM (creating X and y data)
def create_dataset(data, time_step=1):
    X, y = [], []
    for i in range(len(data) - time_step):
        X.append(data[i:(i + time_step), 0])
        y.append(data[i + time_step, 0])
    return np.array(X), np.array(y)

# Reshape data into the form that can be used by LSTM (samples, time steps, features)
time_step = 30 # Use the past 30 days to predict the next day's created issues
X_train, y_train = create_dataset(train_data, time_step)
X_test, y_test = create_dataset(test_data, time_step)

X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)

# Build the LSTM model
model_created_issues = Sequential()

```

```

model_created_issues.add(LSTM(units=50, return_sequences=False,
input_shape=(X_train.shape[1], 1)))
model_created_issues.add(Dense(units=1))

# Compile the model
model_created_issues.compile(optimizer='adam',
loss='mean_squared_error')

# Train the model
model_created_issues.fit(X_train, y_train, epochs=50, batch_size=16,
validation_data=(X_test, y_test), verbose=1)

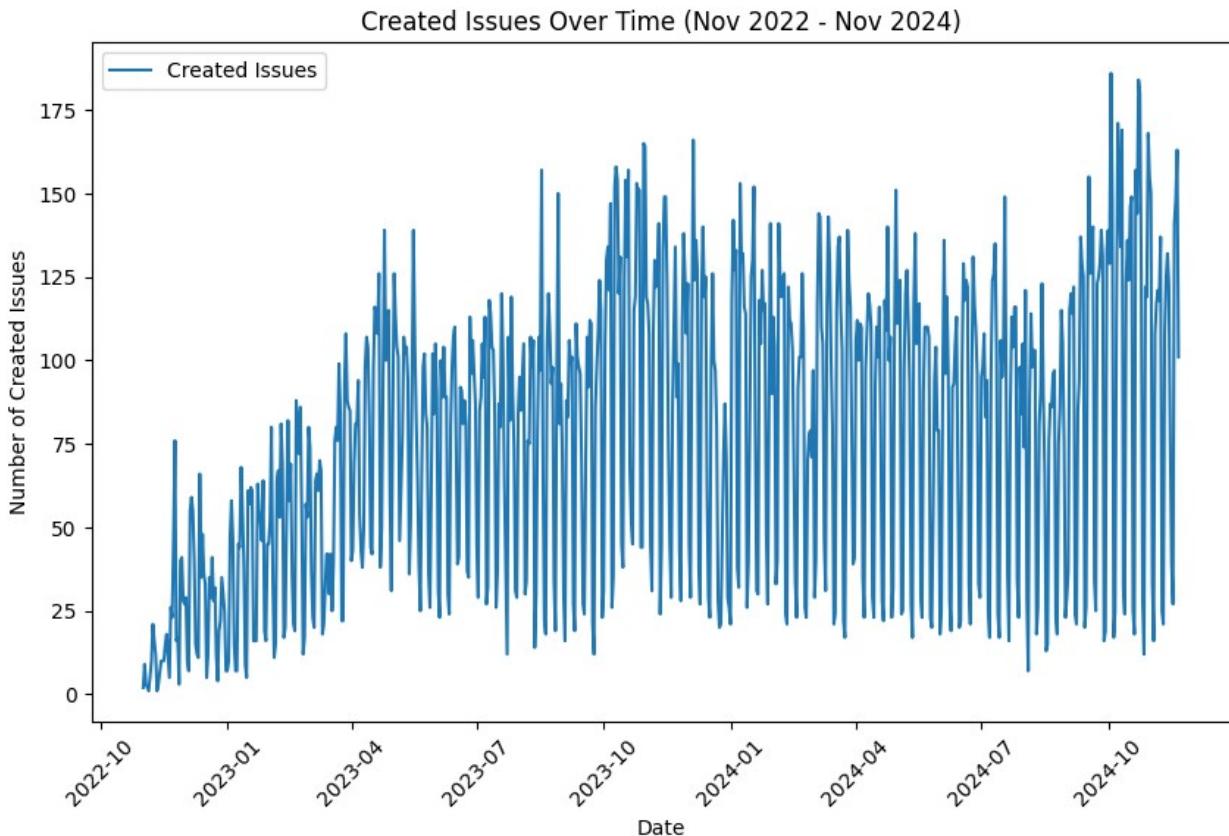
# Predict the created issues on the test set
predicted_created_issues = model_created_issues.predict(X_test)

# Inverse transform the predicted values to get them back to original scale
predicted_created_issues =
scaler_created_issues.inverse_transform(predicted_created_issues)

# Inverse transform the actual values to get them back to original scale
y_test_actual =
scaler_created_issues.inverse_transform(y_test.reshape(-1, 1))

# Plot the original vs predicted values
plt.figure(figsize=(10, 6))
plt.plot(df_created_daily['date'][len(y_test_actual):],
y_test_actual, label='Actual Created Issues')
plt.plot(df_created_daily['date'][len(predicted_created_issues):],
predicted_created_issues, label='Predicted Created Issues',
linestyle='--')
plt.xlabel('Date')
plt.ylabel('Number of Created Issues')
plt.title('Created Issues: Actual vs Predicted (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.savefig("forecasting/Tensorflow_LSTM/issues_created_over_time.png")
plt.show()

```



```

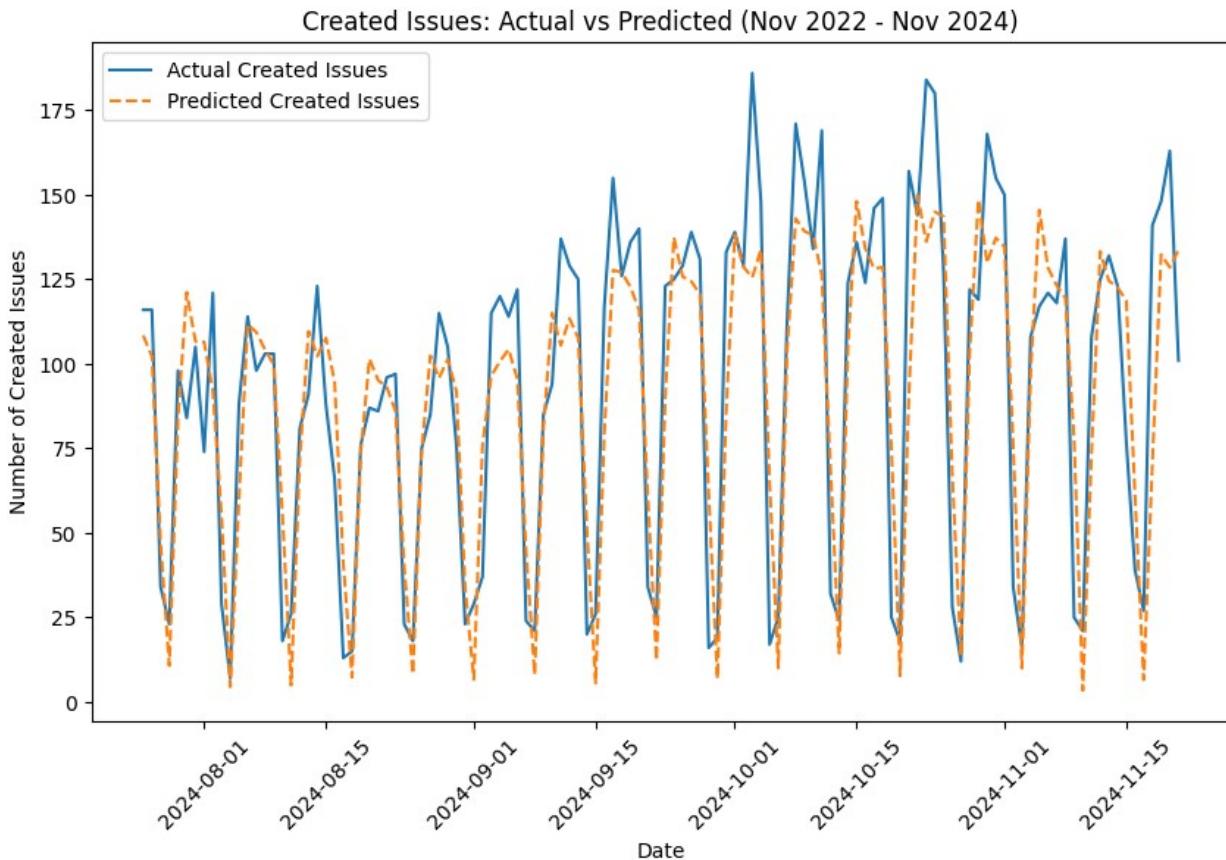
Epoch 1/50
36/36 ━━━━━━━━━━ 3s 17ms/step - loss: 0.1625 - val_loss:
0.0830
Epoch 2/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0436 - val_loss:
0.0721
Epoch 3/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0409 - val_loss:
0.0717
Epoch 4/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0400 - val_loss:
0.0700
Epoch 5/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0400 - val_loss:
0.0694
Epoch 6/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0400 - val_loss:
0.0664
Epoch 7/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0409 - val_loss:
0.0646
Epoch 8/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0361 - val_loss:

```

```
0.0653
Epoch 9/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0325 - val_loss:
0.0669
Epoch 10/50
36/36 ━━━━━━━━ 1s 9ms/step - loss: 0.0323 - val_loss:
0.0342
Epoch 11/50
36/36 ━━━━━━ 0s 10ms/step - loss: 0.0236 - val_loss:
0.0311
Epoch 12/50
36/36 ━━━━ 0s 10ms/step - loss: 0.0194 - val_loss:
0.0287
Epoch 13/50
36/36 ━━ 0s 9ms/step - loss: 0.0211 - val_loss:
0.0284
Epoch 14/50
36/36 ━ 0s 10ms/step - loss: 0.0173 - val_loss:
0.0297
Epoch 15/50
36/36 0s 10ms/step - loss: 0.0191 - val_loss:
0.0315
Epoch 16/50
36/36 0s 10ms/step - loss: 0.0172 - val_loss:
0.0284
Epoch 17/50
36/36 0s 10ms/step - loss: 0.0177 - val_loss:
0.0292
Epoch 18/50
36/36 0s 11ms/step - loss: 0.0171 - val_loss:
0.0265
Epoch 19/50
36/36 0s 11ms/step - loss: 0.0169 - val_loss:
0.0270
Epoch 20/50
36/36 0s 10ms/step - loss: 0.0143 - val_loss:
0.0253
Epoch 21/50
36/36 0s 10ms/step - loss: 0.0162 - val_loss:
0.0273
Epoch 22/50
36/36 0s 10ms/step - loss: 0.0160 - val_loss:
0.0235
Epoch 23/50
36/36 0s 10ms/step - loss: 0.0145 - val_loss:
0.0230
Epoch 24/50
36/36 0s 12ms/step - loss: 0.0150 - val_loss:
0.0269
```

```
Epoch 25/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0178 - val_loss:
0.0243
Epoch 26/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0130 - val_loss:
0.0279
Epoch 27/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0132 - val_loss:
0.0176
Epoch 28/50
36/36 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0108 - val_loss:
0.0203
Epoch 29/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0110 - val_loss:
0.0170
Epoch 30/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0107 - val_loss:
0.0202
Epoch 31/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0105 - val_loss:
0.0167
Epoch 32/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0107 - val_loss:
0.0147
Epoch 33/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0100 - val_loss:
0.0216
Epoch 34/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0102 - val_loss:
0.0212
Epoch 35/50
36/36 ━━━━━━━━━━ 0s 8ms/step - loss: 0.0109 - val_loss:
0.0223
Epoch 36/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0089 - val_loss:
0.0182
Epoch 37/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0103 - val_loss:
0.0182
Epoch 38/50
36/36 ━━━━━━━━━━ 0s 8ms/step - loss: 0.0103 - val_loss:
0.0201
Epoch 39/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0097 - val_loss:
0.0247
Epoch 40/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0105 - val_loss:
0.0161
Epoch 41/50
```

```
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0093 - val_loss:  
0.0173  
Epoch 42/50  
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0096 - val_loss:  
0.0185  
Epoch 43/50  
36/36 ━━━━━━━━ 0s 9ms/step - loss: 0.0086 - val_loss:  
0.0159  
Epoch 44/50  
36/36 ━━━━━━━━ 0s 10ms/step - loss: 0.0095 - val_loss:  
0.0219  
Epoch 45/50  
36/36 ━━━━━━━━ 0s 11ms/step - loss: 0.0092 - val_loss:  
0.0211  
Epoch 46/50  
36/36 ━━━━━━━━ 0s 9ms/step - loss: 0.0088 - val_loss:  
0.0159  
Epoch 47/50  
36/36 ━━━━━━━━ 0s 9ms/step - loss: 0.0099 - val_loss:  
0.0241  
Epoch 48/50  
36/36 ━━━━━━━━ 0s 8ms/step - loss: 0.0090 - val_loss:  
0.0158  
Epoch 49/50  
36/36 ━━━━━━━━ 0s 9ms/step - loss: 0.0079 - val_loss:  
0.0209  
Epoch 50/50  
36/36 ━━━━━━━━ 0s 9ms/step - loss: 0.0089 - val_loss:  
0.0183  
1/4 ━━━━━━━━ 0s 181ms/stepWARNING:tensorflow:5 out of the  
last 7 calls to <function  
TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed at 0x0000021E65BAAB00> triggered tf.function retracing.  
Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce_retracing=True option that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling\_retracing and https://www.tensorflow.org/api\_docs/python/tf/function for more details.  
4/4 ━━━━━━━━ 0s 57ms/step
```



```
#Add your code for requirement 8.5 in this cell
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter the data for 'issue' type and 'closed' status
df_closed_issues = df_detailed[df_detailed['type'] == 'issue']
df_closed_issues = df_closed_issues[df_closed_issues['state'] == 'closed']

# Convert 'created_at' to datetime
df_closed_issues['created_at'] =
pd.to_datetime(df_closed_issues['created_at'])

# Filter data between November 2022 and November 2024
```

```

start_date = '2022-11-01'
end_date = '2024-11-30'
df_closed_issues = df_closed_issues[(df_closed_issues['created_at'] >=
start_date) & (df_closed_issues['created_at'] <= end_date)]

# Extract the date part (ignoring time) and group by date to count
closed issues
df_closed_issues['date'] = df_closed_issues['created_at'].dt.date
df_closed_daily =
df_closed_issues.groupby('date').size().reset_index(name='closed_issue
_count')

# Plot the original closed issues over time (for visualization)
plt.figure(figsize=(10, 6))
plt.plot(df_closed_daily['date'],
df_closed_daily['closed_issue_count'], label='Closed Issues')
plt.xlabel('Date')
plt.ylabel('Number of Closed Issues')
plt.title('Closed Issues Over Time (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.show()

# Prepare the time series data for LSTM
closed_issues_values =
df_closed_daily['closed_issue_count'].values.reshape(-1, 1)

# Normalize the data (MinMaxScaler)
scaler_closed_issues = MinMaxScaler(feature_range=(0, 1))
closed_issues_scaled =
scaler_closed_issues.fit_transform(closed_issues_values)

# Split data into training and testing sets (80% train, 20% test)
train_size = int(len(closed_issues_scaled) * 0.8)
train_data, test_data = closed_issues_scaled[:train_size],
closed_issues_scaled[train_size:]

# Prepare the data for LSTM (creating X and y data)
def create_dataset(data, time_step=1):
    X, y = [], []
    for i in range(len(data) - time_step):
        X.append(data[i:(i + time_step), 0])
        y.append(data[i + time_step, 0])
    return np.array(X), np.array(y)

# Reshape data into the form that can be used by LSTM (samples, time
steps, features)
time_step = 30 # Use the past 30 days to predict the next day's
closed issues
X_train, y_train = create_dataset(train_data, time_step)

```

```

X_test, y_test = create_dataset(test_data, time_step)

X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)

# Build the LSTM model for closed issues
model_closed_issues = Sequential()
model_closed_issues.add(LSTM(units=50, return_sequences=False,
input_shape=(X_train.shape[1], 1)))
model_closed_issues.add(Dense(units=1))

# Compile the model
model_closed_issues.compile(optimizer='adam',
loss='mean_squared_error')

# Train the model
model_closed_issues.fit(X_train, y_train, epochs=50, batch_size=16,
validation_data=(X_test, y_test), verbose=1)

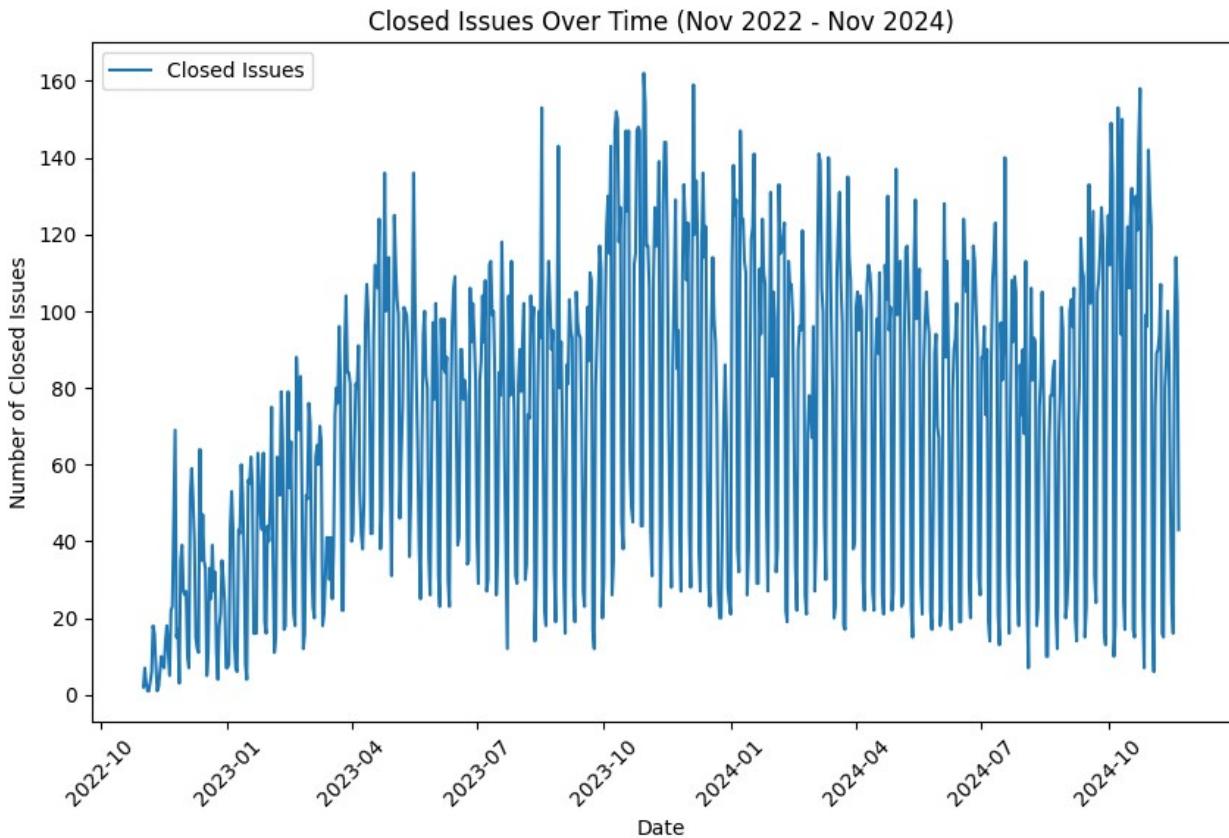
# Predict the closed issues on the test set
predicted_closed_issues = model_closed_issues.predict(X_test)

# Inverse transform the predicted values to get them back to original scale
predicted_closed_issues =
scaler_closed_issues.inverse_transform(predicted_closed_issues)

# Inverse transform the actual values to get them back to original scale
y_test_actual =
scaler_closed_issues.inverse_transform(y_test.reshape(-1, 1))

# Plot the original vs predicted values
plt.figure(figsize=(10, 6))
plt.plot(df_closed_daily['date'][-len(y_test_actual):], y_test_actual,
label='Actual Closed Issues')
plt.plot(df_closed_daily['date'][-len(predicted_closed_issues):],
predicted_closed_issues, label='Predicted Closed Issues',
linestyle='--')
plt.xlabel('Date')
plt.ylabel('Number of Closed Issues')
plt.title('Closed Issues: Actual vs Predicted (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.savefig("forecasting/Tensorflow_LSTM/closed_issues_forecast.png")
plt.show()

```

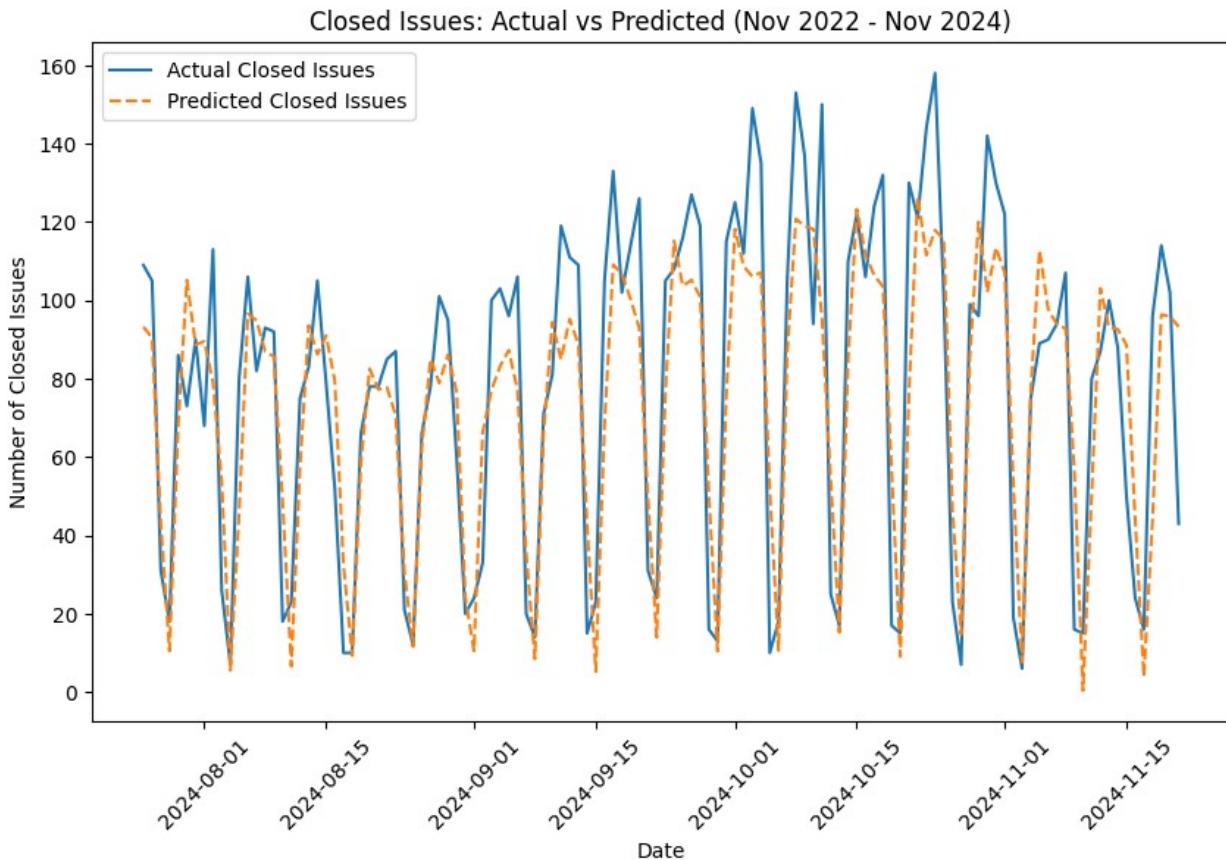


```
Epoch 1/50
36/36 ━━━━━━━━━━ 3s 18ms/step - loss: 0.0981 - val_loss:
0.0733
Epoch 2/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0552 - val_loss:
0.0778
Epoch 3/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0522 - val_loss:
0.0727
Epoch 4/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0510 - val_loss:
0.0711
Epoch 5/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0484 - val_loss:
0.0698
Epoch 6/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0478 - val_loss:
0.0682
Epoch 7/50
36/36 ━━━━━━━━━━ 0s 13ms/step - loss: 0.0446 - val_loss:
0.0651
Epoch 8/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0446 - val_loss:
```

```
0.0624
Epoch 9/50
36/36 ━━━━━━━━━━ 1s 13ms/step - loss: 0.0439 - val_loss:
0.0535
Epoch 10/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0347 - val_loss:
0.0372
Epoch 11/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0252 - val_loss:
0.0388
Epoch 12/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0273 - val_loss:
0.0323
Epoch 13/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0240 - val_loss:
0.0311
Epoch 14/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0222 - val_loss:
0.0306
Epoch 15/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0198 - val_loss:
0.0303
Epoch 16/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0219 - val_loss:
0.0298
Epoch 17/50
36/36 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0192 - val_loss:
0.0322
Epoch 18/50
36/36 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0228 - val_loss:
0.0321
Epoch 19/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0201 - val_loss:
0.0282
Epoch 20/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0195 - val_loss:
0.0279
Epoch 21/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0204 - val_loss:
0.0271
Epoch 22/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0188 - val_loss:
0.0266
Epoch 23/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0205 - val_loss:
0.0286
Epoch 24/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0184 - val_loss:
0.0274
```

```
Epoch 25/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0178 - val_loss:
0.0243
Epoch 26/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0175 - val_loss:
0.0250
Epoch 27/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0171 - val_loss:
0.0197
Epoch 28/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0144 - val_loss:
0.0223
Epoch 29/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0115 - val_loss:
0.0258
Epoch 30/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0143 - val_loss:
0.0232
Epoch 31/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0130 - val_loss:
0.0264
Epoch 32/50
36/36 ━━━━━━━━━━ 0s 8ms/step - loss: 0.0117 - val_loss:
0.0232
Epoch 33/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0115 - val_loss:
0.0220
Epoch 34/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0117 - val_loss:
0.0269
Epoch 35/50
36/36 ━━━━━━━━━━ 0s 8ms/step - loss: 0.0119 - val_loss:
0.0233
Epoch 36/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0127 - val_loss:
0.0207
Epoch 37/50
36/36 ━━━━━━━━━━ 0s 8ms/step - loss: 0.0111 - val_loss:
0.0168
Epoch 38/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0116 - val_loss:
0.0277
Epoch 39/50
36/36 ━━━━━━━━━━ 1s 13ms/step - loss: 0.0119 - val_loss:
0.0254
Epoch 40/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0125 - val_loss:
0.0192
Epoch 41/50
```

```
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0116 - val_loss:  
0.0177  
Epoch 42/50  
36/36 ━━━━━━ 1s 8ms/step - loss: 0.0111 - val_loss:  
0.0180  
Epoch 43/50  
36/36 ━━━━━━ 0s 10ms/step - loss: 0.0101 - val_loss:  
0.0181  
Epoch 44/50  
36/36 ━━━━━━ 0s 9ms/step - loss: 0.0124 - val_loss:  
0.0212  
Epoch 45/50  
36/36 ━━━━━━ 0s 11ms/step - loss: 0.0105 - val_loss:  
0.0158  
Epoch 46/50  
36/36 ━━━━━━ 0s 12ms/step - loss: 0.0130 - val_loss:  
0.0168  
Epoch 47/50  
36/36 ━━━━━━ 0s 12ms/step - loss: 0.0106 - val_loss:  
0.0183  
Epoch 48/50  
36/36 ━━━━━━ 0s 12ms/step - loss: 0.0115 - val_loss:  
0.0158  
Epoch 49/50  
36/36 ━━━━━━ 0s 12ms/step - loss: 0.0109 - val_loss:  
0.0226  
Epoch 50/50  
36/36 ━━━━━━ 0s 11ms/step - loss: 0.0132 - val_loss:  
0.0197  
WARNING:tensorflow:6 out of the last 8 calls to <function  
TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed  
at 0x0000021E5F8F6A70> triggered tf.function retracing.  
Tracing is expensive and the excessive number of tracings could be due  
to (1) creating @tf.function repeatedly in a loop, (2) passing tensors  
with different shapes, (3) passing Python objects instead of tensors.  
For (1), please define your @tf.function outside of the loop. For (2),  
@tf.function has reduce_retracing=True option that can avoid  
unnecessary retracing. For (3), please refer to  
https://www.tensorflow.org/guide/function#controlling\_retracing and  
https://www.tensorflow.org/api\_docs/python/tf/function for more  
details.  
4/4 ━━━━━━ 0s 74ms/step
```



```
#Add your code for requirement 8.6 in this cell
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter the data for 'pull_request' type and 'closed' status
# (assuming 'closed' pull requests)
df_pulls = df_detailed[df_detailed['type'] == 'pull_request']
df_pulls = df_pulls[df_pulls['state'] == 'closed']

# Convert 'created_at' to datetime
df_pulls['created_at'] = pd.to_datetime(df_pulls['created_at'])

# Filter data between November 2022 and November 2024
start_date = '2022-11-01'
```

```

end_date = '2024-11-30'
df_pulls = df_pulls[(df_pulls['created_at'] >= start_date) &
(df_pulls['created_at'] <= end_date)]

# Extract the date part (ignoring time) and group by date to count
pull requests
df_pulls['date'] = df_pulls['created_at'].dt.date
df_pulls_daily =
df_pulls.groupby('date').size().reset_index(name='pull_request_count')

# Plot the original pull requests over time (for visualization)
plt.figure(figsize=(10, 6))
plt.plot(df_pulls_daily['date'], df_pulls_daily['pull_request_count'],
label='Pull Requests')
plt.xlabel('Date')
plt.ylabel('Number of Pull Requests')
plt.title('Pull Requests Over Time (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.show()

# Prepare the time series data for LSTM
pulls_values = df_pulls_daily['pull_request_count'].values.reshape(-1,
1)

# Normalize the data (MinMaxScaler)
scaler_pulls = MinMaxScaler(feature_range=(0, 1))
pulls_scaled = scaler_pulls.fit_transform(pulls_values)

# Split data into training and testing sets (80% train, 20% test)
train_size = int(len(pulls_scaled) * 0.8)
train_data, test_data = pulls_scaled[:train_size],
pulls_scaled[train_size:]

# Prepare the data for LSTM (creating X and y data)
def create_dataset(data, time_step=1):
    X, y = [], []
    for i in range(len(data) - time_step):
        X.append(data[i:(i + time_step), 0])
        y.append(data[i + time_step, 0])
    return np.array(X), np.array(y)

# Reshape data into the form that can be used by LSTM (samples, time
steps, features)
time_step = 30 # Use the past 30 days to predict the next day's pull
requests
X_train, y_train = create_dataset(train_data, time_step)
X_test, y_test = create_dataset(test_data, time_step)

X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)

```

```

X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)

# Build the LSTM model for pull requests
model_pulls = Sequential()
model_pulls.add(LSTM(units=50, return_sequences=False,
input_shape=(X_train.shape[1], 1)))
model_pulls.add(Dense(units=1))

# Compile the model
model_pulls.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
model_pulls.fit(X_train, y_train, epochs=50, batch_size=16,
validation_data=(X_test, y_test), verbose=1)

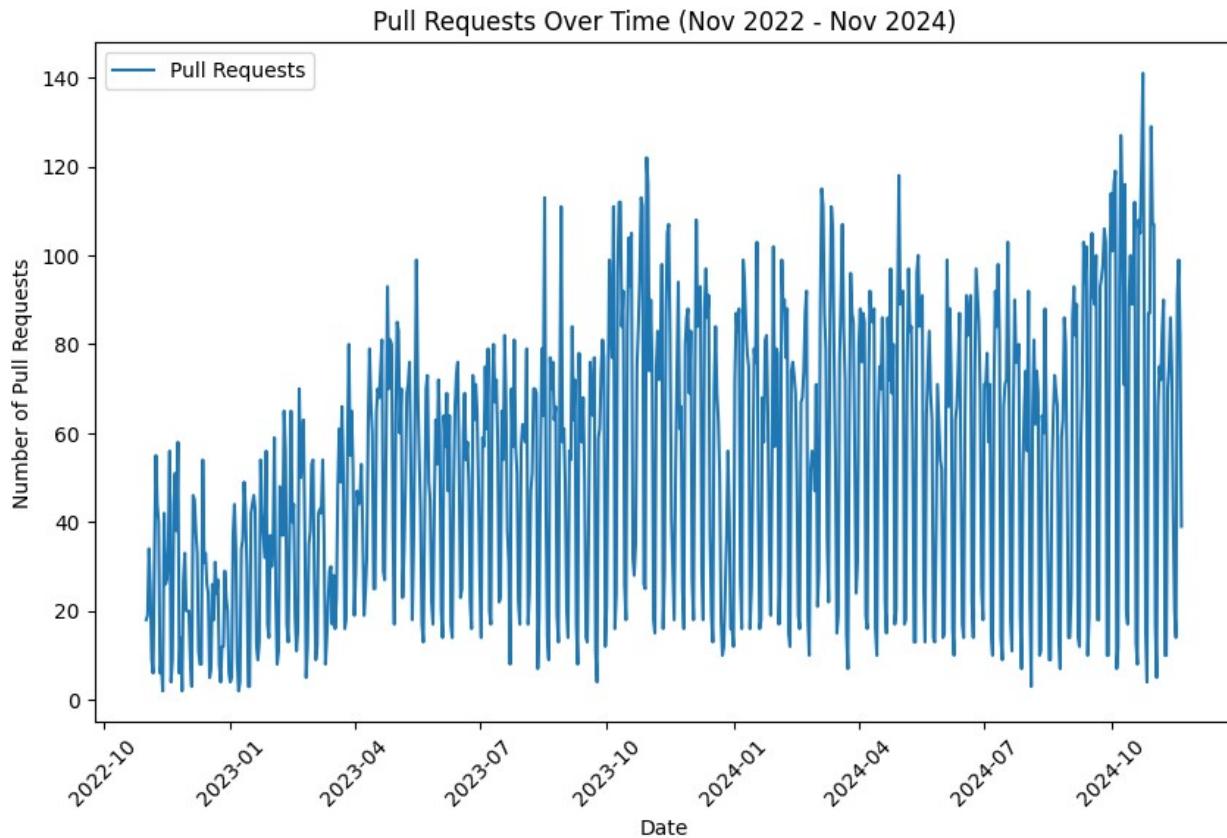
# Predict the pull requests on the test set
predicted_pulls = model_pulls.predict(X_test)

# Inverse transform the predicted values to get them back to original scale
predicted_pulls = scaler_pulls.inverse_transform(predicted_pulls)

# Inverse transform the actual values to get them back to original scale
y_test_actual = scaler_pulls.inverse_transform(y_test.reshape(-1, 1))

# Plot the original vs predicted values
plt.figure(figsize=(10, 6))
plt.plot(df_pulls_daily['date'][-len(y_test_actual):], y_test_actual,
label='Actual Pull Requests')
plt.plot(df_pulls_daily['date'][-len(predicted_pulls):],
predicted_pulls, label='Predicted Pull Requests', linestyle='--')
plt.xlabel('Date')
plt.ylabel('Number of Pull Requests')
plt.title('Pull Requests: Actual vs Predicted (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.savefig("forecasting/Tensorflow_LSTM/pull_forecast.png")
plt.show()

```

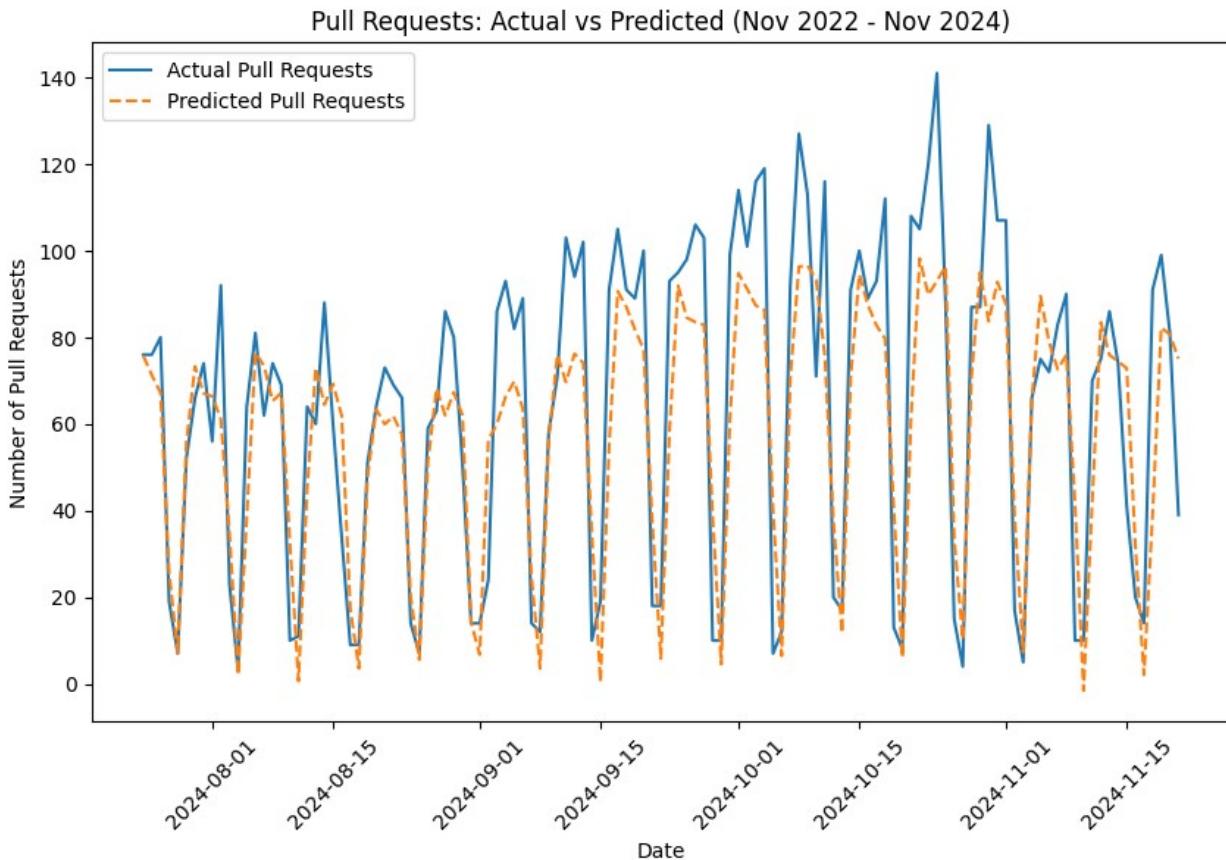


```
Epoch 1/50
36/36 ━━━━━━━━━━ 3s 23ms/step - loss: 0.0857 - val_loss:
0.0725
Epoch 2/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0402 - val_loss:
0.0712
Epoch 3/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0391 - val_loss:
0.0717
Epoch 4/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0411 - val_loss:
0.0692
Epoch 5/50
36/36 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0374 - val_loss:
0.0664
Epoch 6/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0376 - val_loss:
0.0645
Epoch 7/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0330 - val_loss:
0.0672
Epoch 8/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0369 - val_loss:
```

```
0.0662
Epoch 9/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0334 - val_loss:
0.0548
Epoch 10/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0296 - val_loss:
0.0477
Epoch 11/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0263 - val_loss:
0.0351
Epoch 12/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0243 - val_loss:
0.0340
Epoch 13/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0223 - val_loss:
0.0346
Epoch 14/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0194 - val_loss:
0.0325
Epoch 15/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0166 - val_loss:
0.0313
Epoch 16/50
36/36 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0182 - val_loss:
0.0312
Epoch 17/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0172 - val_loss:
0.0311
Epoch 18/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0196 - val_loss:
0.0336
Epoch 19/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0184 - val_loss:
0.0309
Epoch 20/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0172 - val_loss:
0.0311
Epoch 21/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0170 - val_loss:
0.0368
Epoch 22/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0172 - val_loss:
0.0277
Epoch 23/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0159 - val_loss:
0.0230
Epoch 24/50
36/36 ━━━━━━━━━━ 0s 13ms/step - loss: 0.0162 - val_loss:
0.0205
```

```
Epoch 25/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0137 - val_loss:
0.0190
Epoch 26/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0128 - val_loss:
0.0213
Epoch 27/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0108 - val_loss:
0.0209
Epoch 28/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0117 - val_loss:
0.0195
Epoch 29/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0112 - val_loss:
0.0189
Epoch 30/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0098 - val_loss:
0.0171
Epoch 31/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0107 - val_loss:
0.0195
Epoch 32/50
36/36 ━━━━━━━━━━ 0s 8ms/step - loss: 0.0097 - val_loss:
0.0187
Epoch 33/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0096 - val_loss:
0.0323
Epoch 34/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0096 - val_loss:
0.0183
Epoch 35/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0092 - val_loss:
0.0197
Epoch 36/50
36/36 ━━━━━━━━━━ 1s 12ms/step - loss: 0.0095 - val_loss:
0.0255
Epoch 37/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0115 - val_loss:
0.0244
Epoch 38/50
36/36 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0111 - val_loss:
0.0211
Epoch 39/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0094 - val_loss:
0.0174
Epoch 40/50
36/36 ━━━━━━━━━━ 0s 13ms/step - loss: 0.0112 - val_loss:
0.0162
Epoch 41/50
```

```
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0105 - val_loss:  
0.0251  
Epoch 42/50  
36/36 ━━━━━━━━ 0s 11ms/step - loss: 0.0113 - val_loss:  
0.0164  
Epoch 43/50  
36/36 ━━━━━━ 0s 11ms/step - loss: 0.0096 - val_loss:  
0.0175  
Epoch 44/50  
36/36 ━━━━ 1s 20ms/step - loss: 0.0109 - val_loss:  
0.0187  
Epoch 45/50  
36/36 ━━━━ 0s 12ms/step - loss: 0.0102 - val_loss:  
0.0198  
Epoch 46/50  
36/36 ━━━━ 0s 11ms/step - loss: 0.0102 - val_loss:  
0.0185  
Epoch 47/50  
36/36 ━━━━ 0s 10ms/step - loss: 0.0102 - val_loss:  
0.0221  
Epoch 48/50  
36/36 ━━━━ 0s 8ms/step - loss: 0.0104 - val_loss:  
0.0172  
Epoch 49/50  
36/36 ━━━━ 0s 10ms/step - loss: 0.0096 - val_loss:  
0.0189  
Epoch 50/50  
36/36 ━━━━ 0s 12ms/step - loss: 0.0105 - val_loss:  
0.0204  
4/4 ━━━━ 0s 78ms/step
```



```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler
from datetime import datetime

# Load the data
df_commits = pd.read_csv("github_all_data.csv")

# Extract commit date (assuming the 'created_at' column contains
# commit timestamp)
df_commits['created_at'] = pd.to_datetime(df_commits['created_at'])

# Filter commits from November 2022 to November 2024 (adjust dates if
# needed)
start_date = '2022-11-01'
end_date = '2024-11-30'
df_commits = df_commits[(df_commits['created_at'] >= start_date) &
(df_commits['created_at'] <= end_date)]

# Group by date and count the number of commits per day

```

```

df_commits['date'] = df_commits['created_at'].dt.date
df_commits_daily =
df_commits.groupby('date').size().reset_index(name='commit_count')

# Plot the commits over time
plt.figure(figsize=(10, 6))
plt.plot(df_commits_daily['date'], df_commits_daily['commit_count'],
label='Commit Count')
plt.xlabel('Date')
plt.ylabel('Number of Commits')
plt.title('Commits Over Time (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.show()

# Prepare the time series data for LSTM
commits_values = df_commits_daily['commit_count'].values.reshape(-1,
1)

# Normalize the data
scaler_commits = MinMaxScaler(feature_range=(0, 1))
commits_scaled = scaler_commits.fit_transform(commits_values)

# Split the data into training and testing sets (80% train, 20% test)
train_size = int(len(commits_scaled) * 0.8)
train_data, test_data = commits_scaled[:train_size],
commits_scaled[train_size:]

# Create the dataset for LSTM (X, y)
def create_dataset(data, time_step=1):
    X, y = [], []
    for i in range(len(data) - time_step):
        X.append(data[i:(i + time_step), 0])
        y.append(data[i + time_step, 0])
    return np.array(X), np.array(y)

# Reshape data into the form that can be used by LSTM (samples, time steps, features)
time_step = 30 # Use the past 30 days to predict the next day's
commits
X_train, y_train = create_dataset(train_data, time_step)
X_test, y_test = create_dataset(test_data, time_step)

X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)

# Build the LSTM model
model_commits = Sequential()
model_commits.add(LSTM(units=50, return_sequences=False,
input_shape=(X_train.shape[1], 1)))

```

```

model_commits.add(Dense(units=1))

# Compile the model
model_commits.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
model_commits.fit(X_train, y_train, epochs=50, batch_size=16,
validation_data=(X_test, y_test), verbose=1)

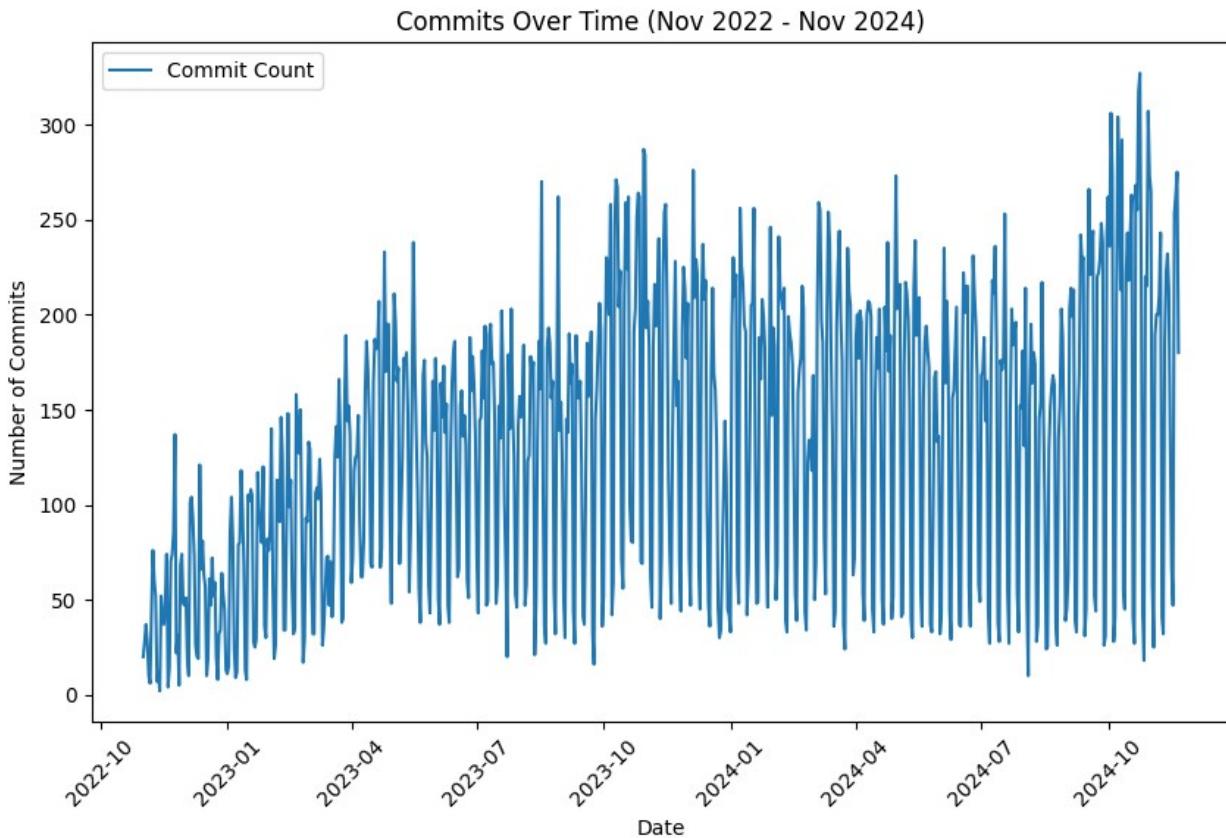
# Predict the commits on the test set
predicted_commits = model_commits.predict(X_test)

# Inverse transform the predicted values to get them back to original scale
predicted_commits =
scaler_commits.inverse_transform(predicted_commits)

# Inverse transform the actual values to get them back to original scale
y_test_actual = scaler_commits.inverse_transform(y_test.reshape(-1,
1))

# Plot the actual vs predicted commits
plt.figure(figsize=(10, 6))
plt.plot(df_commits_daily['date'][-len(y_test_actual):],
y_test_actual, label='Actual Commits')
plt.plot(df_commits_daily['date'][-len(predicted_commits):],
predicted_commits, label='Predicted Commits', linestyle='--')
plt.xlabel('Date')
plt.ylabel('Number of Commits')
plt.title('Commits: Actual vs Predicted (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.savefig("forecasting/Tensorflow_LSTM/commits_forecast.png")
plt.show()

```



```

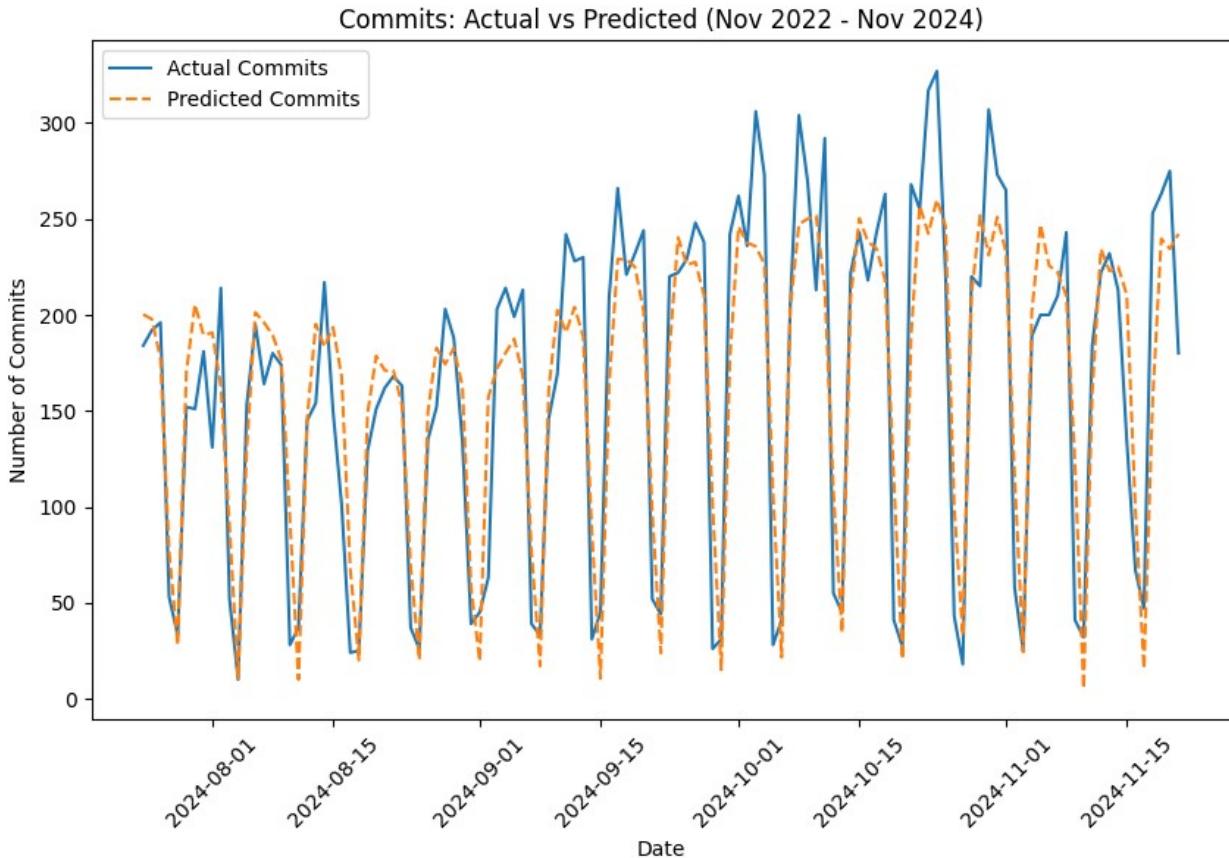
Epoch 1/50
36/36 ━━━━━━━━━━ 3s 29ms/step - loss: 0.1036 - val_loss:
0.0745
Epoch 2/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0428 - val_loss:
0.0761
Epoch 3/50
36/36 ━━━━━━━━━━ 1s 13ms/step - loss: 0.0405 - val_loss:
0.0710
Epoch 4/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0437 - val_loss:
0.0721
Epoch 5/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0417 - val_loss:
0.0755
Epoch 6/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0394 - val_loss:
0.0689
Epoch 7/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0386 - val_loss:
0.0664
Epoch 8/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0381 - val_loss:

```

```
0.0656
Epoch 9/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0355 - val_loss:
0.0590
Epoch 10/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0310 - val_loss:
0.0359
Epoch 11/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0219 - val_loss:
0.0396
Epoch 12/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0189 - val_loss:
0.0332
Epoch 13/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0208 - val_loss:
0.0380
Epoch 14/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0184 - val_loss:
0.0346
Epoch 15/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0189 - val_loss:
0.0349
Epoch 16/50
36/36 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0184 - val_loss:
0.0328
Epoch 17/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0179 - val_loss:
0.0296
Epoch 18/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0192 - val_loss:
0.0301
Epoch 19/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0178 - val_loss:
0.0329
Epoch 20/50
36/36 ━━━━━━━━━━ 0s 8ms/step - loss: 0.0184 - val_loss:
0.0262
Epoch 21/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0175 - val_loss:
0.0262
Epoch 22/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0145 - val_loss:
0.0230
Epoch 23/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0149 - val_loss:
0.0217
Epoch 24/50
36/36 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0136 - val_loss:
0.0228
```

```
Epoch 25/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0131 - val_loss:
0.0179
Epoch 26/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0115 - val_loss:
0.0241
Epoch 27/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0115 - val_loss:
0.0154
Epoch 28/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0101 - val_loss:
0.0170
Epoch 29/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0094 - val_loss:
0.0183
Epoch 30/50
36/36 ━━━━━━━━━━ 1s 14ms/step - loss: 0.0111 - val_loss:
0.0157
Epoch 31/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0100 - val_loss:
0.0177
Epoch 32/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0103 - val_loss:
0.0192
Epoch 33/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0090 - val_loss:
0.0135
Epoch 34/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0115 - val_loss:
0.0139
Epoch 35/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0123 - val_loss:
0.0209
Epoch 36/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0086 - val_loss:
0.0216
Epoch 37/50
36/36 ━━━━━━━━━━ 0s 13ms/step - loss: 0.0096 - val_loss:
0.0166
Epoch 38/50
36/36 ━━━━━━━━━━ 1s 13ms/step - loss: 0.0085 - val_loss:
0.0190
Epoch 39/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0091 - val_loss:
0.0211
Epoch 40/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0098 - val_loss:
0.0157
Epoch 41/50
```

```
36/36 ━━━━━━━━━━ 1s 13ms/step - loss: 0.0098 - val_loss:  
0.0190  
Epoch 42/50  
36/36 ━━━━━━━━ 0s 11ms/step - loss: 0.0093 - val_loss:  
0.0177  
Epoch 43/50  
36/36 ━━━━━━━━ 0s 11ms/step - loss: 0.0089 - val_loss:  
0.0185  
Epoch 44/50  
36/36 ━━━━━━━━ 1s 13ms/step - loss: 0.0097 - val_loss:  
0.0216  
Epoch 45/50  
36/36 ━━━━━━━━ 1s 15ms/step - loss: 0.0104 - val_loss:  
0.0166  
Epoch 46/50  
36/36 ━━━━━━━━ 0s 10ms/step - loss: 0.0089 - val_loss:  
0.0166  
Epoch 47/50  
36/36 ━━━━━━━━ 0s 10ms/step - loss: 0.0079 - val_loss:  
0.0155  
Epoch 48/50  
36/36 ━━━━━━━━ 0s 8ms/step - loss: 0.0093 - val_loss:  
0.0138  
Epoch 49/50  
36/36 ━━━━━━━━ 0s 9ms/step - loss: 0.0104 - val_loss:  
0.0144  
Epoch 50/50  
36/36 ━━━━━━━━ 0s 10ms/step - loss: 0.0099 - val_loss:  
0.0146  
4/4 ━━━━━━━━ 0s 56ms/step
```



```
#Add your code for requirement 8.8 in this cell

# Load the data
df_branches = pd.read_csv("github_all_data.csv")

# Assuming there is a 'branch' or 'ref' column to track branch names
# Extract the date (assuming 'created_at' column contains timestamp)
df_branches['created_at'] = pd.to_datetime(df_branches['created_at'])

# Filter data between Nov 2022 and Nov 2024 (adjust as necessary)
df_branches = df_branches[(df_branches['created_at'] >= start_date) &
                           (df_branches['created_at'] <= end_date)]

# Filter for the 'branch' column (assuming it exists)
# Group by date and count the number of branches per day
df_branches_daily =
    df_branches.groupby(df_branches['created_at'].dt.date).size().reset_index(name='branch_count')

# Plot the branches over time
plt.figure(figsize=(10, 6))
plt.plot(df_branches_daily['created_at'],
         df_branches_daily['branch_count'], label='Branch Count')
```

```

plt.xlabel('Date')
plt.ylabel('Number of Branches')
plt.title('Branches Over Time (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.show()

# Prepare the time series data for LSTM
branches_values = df_branches_daily['branch_count'].values.reshape(-1,
1)

# Normalize the data
scaler_branches = MinMaxScaler(feature_range=(0, 1))
branches_scaled = scaler_branches.fit_transform(branches_values)

# Split the data into training and testing sets (80% train, 20% test)
train_size = int(len(branches_scaled) * 0.8)
train_data, test_data = branches_scaled[:train_size],
branches_scaled[train_size:]

# Create the dataset for LSTM (X, y)
def create_dataset(data, time_step=1):
    X, y = [], []
    for i in range(len(data) - time_step):
        X.append(data[i:(i + time_step), 0])
        y.append(data[i + time_step, 0])
    return np.array(X), np.array(y)

# Reshape data into the form that can be used by LSTM (samples, time steps, features)
time_step = 30 # Use the past 30 days to predict the next day's
branches
X_train, y_train = create_dataset(train_data, time_step)
X_test, y_test = create_dataset(test_data, time_step)

X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)

# Build the LSTM model
model_branches = Sequential()
model_branches.add(LSTM(units=50, return_sequences=False,
input_shape=(X_train.shape[1], 1)))
model_branches.add(Dense(units=1))

# Compile the model
model_branches.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
model_branches.fit(X_train, y_train, epochs=50, batch_size=16,
validation_data=(X_test, y_test), verbose=1)

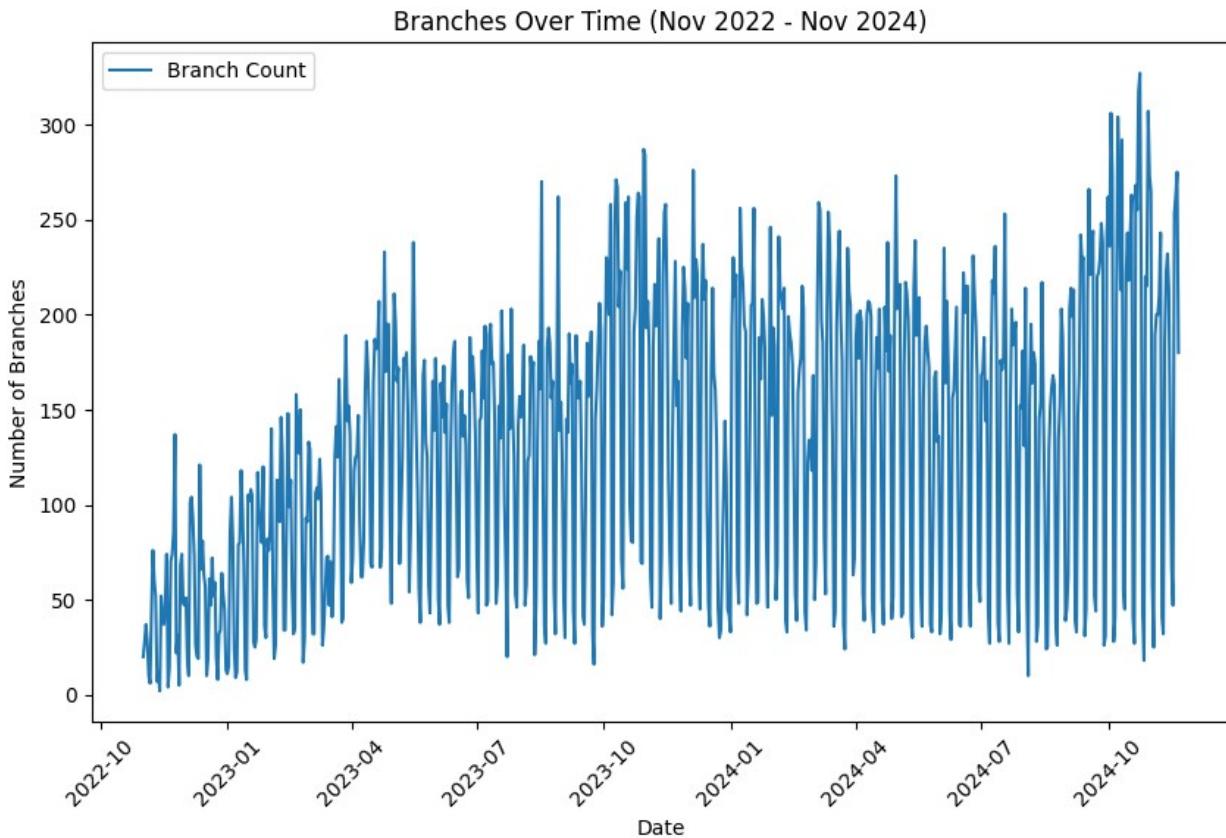
```

```
# Predict the branches on the test set
predicted_branches = model_branches.predict(X_test)

# Inverse transform the predicted values to get them back to original
# scale
predicted_branches =
scaler_branches.inverse_transform(predicted_branches)

# Inverse transform the actual values to get them back to original
# scale
y_test_actual = scaler_branches.inverse_transform(y_test.reshape(-1,
1))

# Plot the actual vs predicted branches
plt.figure(figsize=(10, 6))
plt.plot(df_branches_daily['created_at'][len(y_test_actual):],
y_test_actual, label='Actual Branches')
plt.plot(df_branches_daily['created_at'][len(predicted_branches):],
predicted_branches, label='Predicted Branches', linestyle='--')
plt.xlabel('Date')
plt.ylabel('Number of Branches')
plt.title('Branches: Actual vs Predicted (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.savefig("forecasting/Tensorflow_LSTM/branches_forecast.png")
plt.show()
```



```

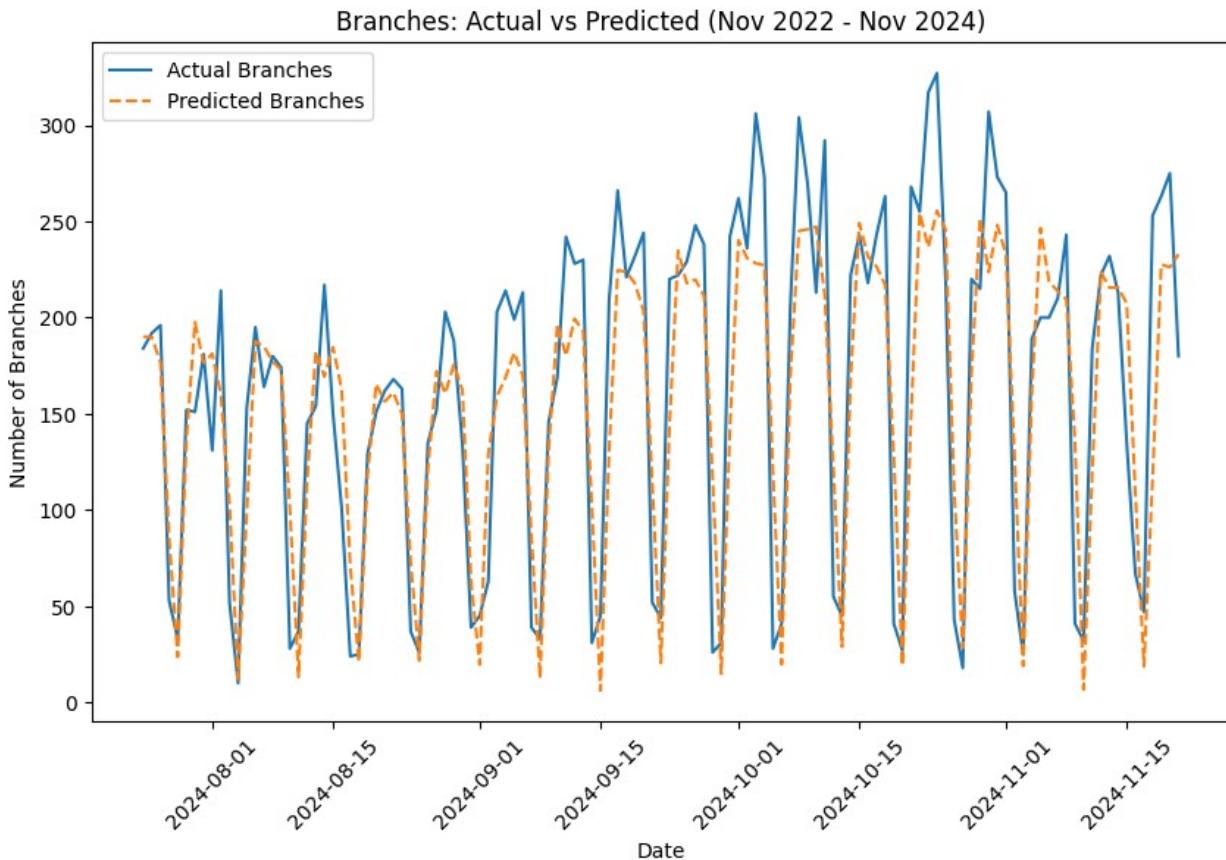
Epoch 1/50
36/36 ━━━━━━━━━━ 3s 19ms/step - loss: 0.0911 - val_loss:
0.0769
Epoch 2/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0405 - val_loss:
0.0755
Epoch 3/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0407 - val_loss:
0.0823
Epoch 4/50
36/36 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0395 - val_loss:
0.0727
Epoch 5/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0403 - val_loss:
0.0712
Epoch 6/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0365 - val_loss:
0.0697
Epoch 7/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0377 - val_loss:
0.0630
Epoch 8/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0375 - val_loss:

```

```
0.0549
Epoch 9/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0293 - val_loss:
0.0415
Epoch 10/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0216 - val_loss:
0.0362
Epoch 11/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0203 - val_loss:
0.0368
Epoch 12/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0204 - val_loss:
0.0307
Epoch 13/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0182 - val_loss:
0.0359
Epoch 14/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0170 - val_loss:
0.0319
Epoch 15/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0193 - val_loss:
0.0296
Epoch 16/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0178 - val_loss:
0.0317
Epoch 17/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0161 - val_loss:
0.0303
Epoch 18/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0164 - val_loss:
0.0302
Epoch 19/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0176 - val_loss:
0.0294
Epoch 20/50
36/36 ━━━━━━━━━━ 0s 13ms/step - loss: 0.0173 - val_loss:
0.0261
Epoch 21/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0165 - val_loss:
0.0284
Epoch 22/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0169 - val_loss:
0.0255
Epoch 23/50
36/36 ━━━━━━━━━━ 0s 13ms/step - loss: 0.0137 - val_loss:
0.0283
Epoch 24/50
36/36 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0152 - val_loss:
0.0221
```

```
Epoch 25/50
36/36 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0119 - val_loss:
0.0254
Epoch 26/50
36/36 ━━━━━━━━ 1s 12ms/step - loss: 0.0095 - val_loss:
0.0299
Epoch 27/50
36/36 ━━━━━━━━ 1s 13ms/step - loss: 0.0115 - val_loss:
0.0245
Epoch 28/50
36/36 ━━━━━━ 0s 11ms/step - loss: 0.0095 - val_loss:
0.0189
Epoch 29/50
36/36 ━━━━━━ 1s 14ms/step - loss: 0.0102 - val_loss:
0.0187
Epoch 30/50
36/36 ━━━━━━ 0s 12ms/step - loss: 0.0098 - val_loss:
0.0246
Epoch 31/50
36/36 ━━━━━━ 1s 13ms/step - loss: 0.0091 - val_loss:
0.0225
Epoch 32/50
36/36 ━━━━━━ 0s 10ms/step - loss: 0.0100 - val_loss:
0.0185
Epoch 33/50
36/36 ━━━━━━ 0s 12ms/step - loss: 0.0087 - val_loss:
0.0161
Epoch 34/50
36/36 ━━━━━━ 1s 14ms/step - loss: 0.0094 - val_loss:
0.0195
Epoch 35/50
36/36 ━━━━━━ 0s 8ms/step - loss: 0.0094 - val_loss:
0.0148
Epoch 36/50
36/36 ━━━━━━ 0s 10ms/step - loss: 0.0094 - val_loss:
0.0152
Epoch 37/50
36/36 ━━━━━━ 0s 13ms/step - loss: 0.0087 - val_loss:
0.0162
Epoch 38/50
36/36 ━━━━━━ 0s 9ms/step - loss: 0.0097 - val_loss:
0.0190
Epoch 39/50
36/36 ━━━━━━ 0s 11ms/step - loss: 0.0106 - val_loss:
0.0173
Epoch 40/50
36/36 ━━━━━━ 0s 10ms/step - loss: 0.0083 - val_loss:
0.0174
Epoch 41/50
```

```
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0090 - val_loss:  
0.0221  
Epoch 42/50  
36/36 ━━━━━━━━ 0s 11ms/step - loss: 0.0086 - val_loss:  
0.0153  
Epoch 43/50  
36/36 ━━━━━━━━ 0s 10ms/step - loss: 0.0094 - val_loss:  
0.0170  
Epoch 44/50  
36/36 ━━━━━━━━ 0s 10ms/step - loss: 0.0095 - val_loss:  
0.0286  
Epoch 45/50  
36/36 ━━━━━━━━ 1s 13ms/step - loss: 0.0118 - val_loss:  
0.0184  
Epoch 46/50  
36/36 ━━━━━━━━ 0s 12ms/step - loss: 0.0089 - val_loss:  
0.0164  
Epoch 47/50  
36/36 ━━━━━━━━ 0s 10ms/step - loss: 0.0093 - val_loss:  
0.0211  
Epoch 48/50  
36/36 ━━━━━━━━ 0s 10ms/step - loss: 0.0094 - val_loss:  
0.0150  
Epoch 49/50  
36/36 ━━━━━━━━ 0s 12ms/step - loss: 0.0093 - val_loss:  
0.0172  
Epoch 50/50  
36/36 ━━━━━━━━ 0s 9ms/step - loss: 0.0092 - val_loss:  
0.0199  
4/4 ━━━━━━━━ 0s 62ms/step
```



```
#Add your code for requirement 8.6 in this cell
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter the data for 'pull_request' type and 'closed' status
# (assuming 'closed' pull requests)
df_pulls = df_detailed[df_detailed['type'] == 'pull_request']
df_pulls = df_pulls[df_pulls['state'] == 'closed']

# Convert 'created_at' to datetime
df_pulls['created_at'] = pd.to_datetime(df_pulls['created_at'])

# Filter data between November 2022 and November 2024
start_date = '2022-11-01'
```

```

end_date = '2024-11-30'
df_pulls = df_pulls[(df_pulls['created_at'] >= start_date) &
(df_pulls['created_at'] <= end_date)]

# Extract the date part (ignoring time) and group by date to count
pull requests
df_pulls['date'] = df_pulls['created_at'].dt.date
df_pulls_daily =
df_pulls.groupby('date').size().reset_index(name='pull_request_count')

# Plot the original pull requests over time (for visualization)
plt.figure(figsize=(10, 6))
plt.plot(df_pulls_daily['date'], df_pulls_daily['pull_request_count'],
label='Pull Requests')
plt.xlabel('Date')
plt.ylabel('Number of Contributors')
plt.title('Number of Contributors Over Time (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.show()

# Prepare the time series data for LSTM
pulls_values = df_pulls_daily['pull_request_count'].values.reshape(-1,
1)

# Normalize the data (MinMaxScaler)
scaler_pulls = MinMaxScaler(feature_range=(0, 1))
pulls_scaled = scaler_pulls.fit_transform(pulls_values)

# Split data into training and testing sets (80% train, 20% test)
train_size = int(len(pulls_scaled) * 0.8)
train_data, test_data = pulls_scaled[:train_size],
pulls_scaled[train_size:]

# Prepare the data for LSTM (creating X and y data)
def create_dataset(data, time_step=1):
    X, y = [], []
    for i in range(len(data) - time_step):
        X.append(data[i:(i + time_step), 0])
        y.append(data[i + time_step, 0])
    return np.array(X), np.array(y)

# Reshape data into the form that can be used by LSTM (samples, time
steps, features)
time_step = 30 # Use the past 30 days to predict the next day's pull
requests
X_train, y_train = create_dataset(train_data, time_step)
X_test, y_test = create_dataset(test_data, time_step)

X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)

```

```

X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)

# Build the LSTM model for pull requests
model_pulls = Sequential()
model_pulls.add(LSTM(units=50, return_sequences=False,
input_shape=(X_train.shape[1], 1)))
model_pulls.add(Dense(units=1))

# Compile the model
model_pulls.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
model_pulls.fit(X_train, y_train, epochs=50, batch_size=16,
validation_data=(X_test, y_test), verbose=1)

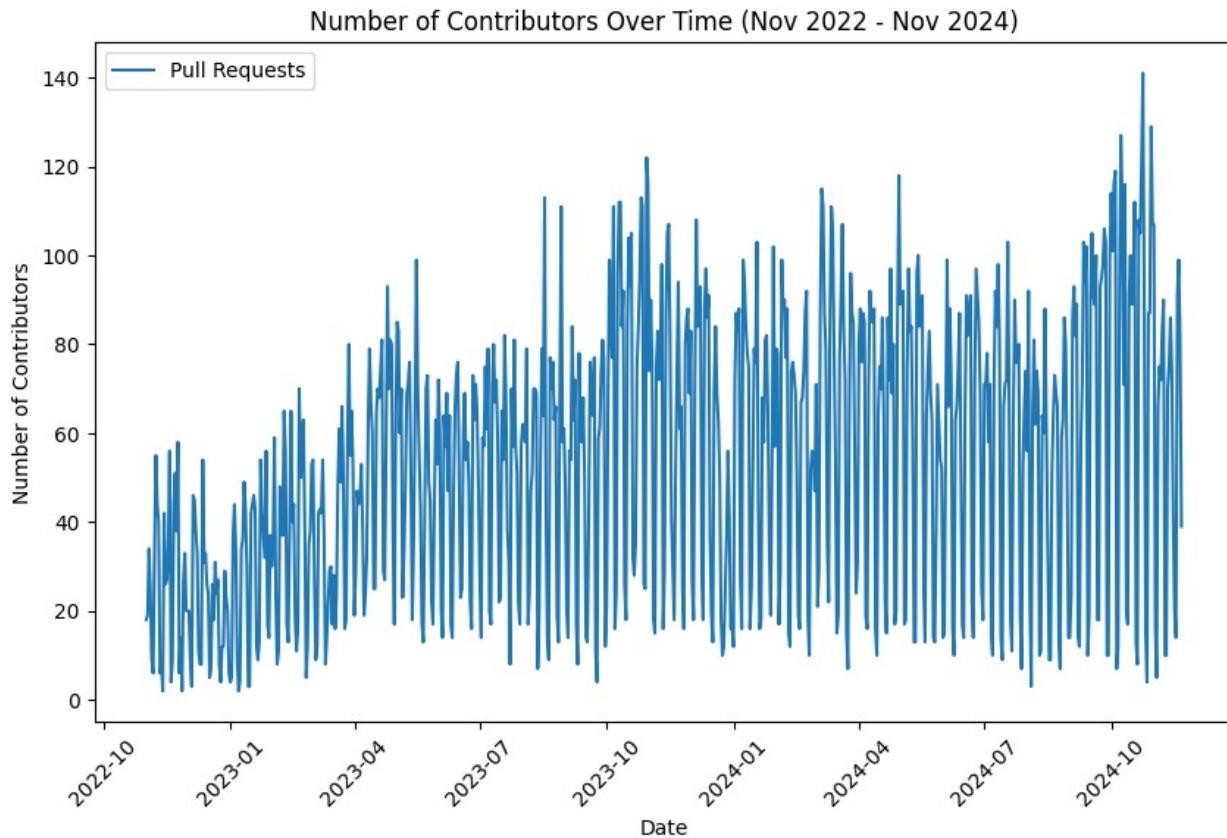
# Predict the pull requests on the test set
predicted_pulls = model_pulls.predict(X_test)

# Inverse transform the predicted values to get them back to original scale
predicted_pulls = scaler_pulls.inverse_transform(predicted_pulls)

# Inverse transform the actual values to get them back to original scale
y_test_actual = scaler_pulls.inverse_transform(y_test.reshape(-1, 1))

# Plot the original vs predicted values
plt.figure(figsize=(10, 6))
plt.plot(df_pulls_daily['date'][-len(y_test_actual):], y_test_actual,
label='Actual Contributors')
plt.plot(df_pulls_daily['date'][-len(predicted_pulls):],
predicted_pulls, label='Predicted Contributors', linestyle='--')
plt.xlabel('Date')
plt.ylabel('Number of Contributors')
plt.title('No. of Contributors: Actual vs Predicted (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.savefig("forecasting/Tensorflow_LSTM/contributors_forecast.png")
plt.show()

```



```

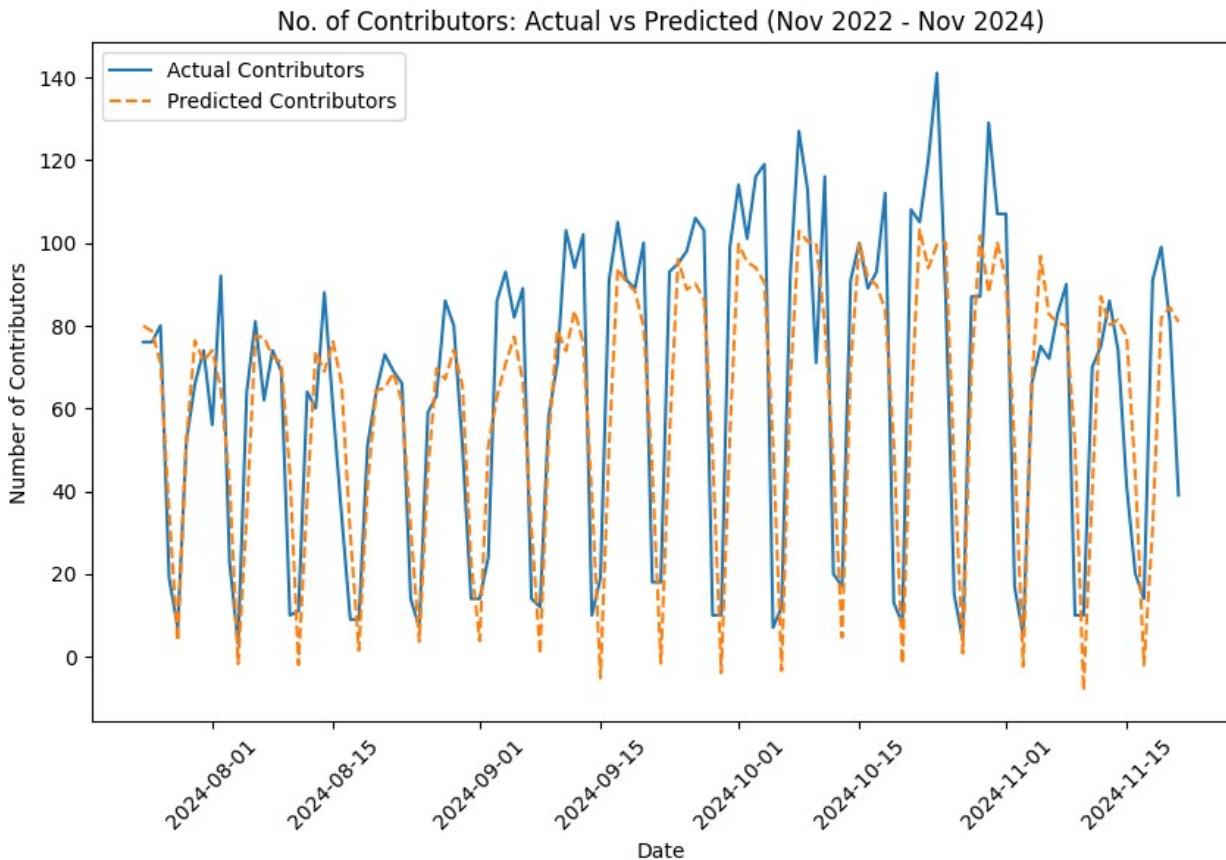
Epoch 1/50
36/36 ━━━━━━━━━━ 3s 17ms/step - loss: 0.0925 - val_loss:
0.0724
Epoch 2/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0406 - val_loss:
0.0718
Epoch 3/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0383 - val_loss:
0.0733
Epoch 4/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0388 - val_loss:
0.0710
Epoch 5/50
36/36 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0407 - val_loss:
0.0692
Epoch 6/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0389 - val_loss:
0.0683
Epoch 7/50
36/36 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0387 - val_loss:
0.0738
Epoch 8/50
36/36 ━━━━━━━━━━ 1s 14ms/step - loss: 0.0370 - val_loss:

```

```
0.0619
Epoch 9/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0348 - val_loss:
0.0534
Epoch 10/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0347 - val_loss:
0.0416
Epoch 11/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0268 - val_loss:
0.0337
Epoch 12/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0228 - val_loss:
0.0343
Epoch 13/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0196 - val_loss:
0.0327
Epoch 14/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0187 - val_loss:
0.0340
Epoch 15/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0216 - val_loss:
0.0313
Epoch 16/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0178 - val_loss:
0.0334
Epoch 17/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0186 - val_loss:
0.0313
Epoch 18/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0181 - val_loss:
0.0295
Epoch 19/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0187 - val_loss:
0.0302
Epoch 20/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0175 - val_loss:
0.0306
Epoch 21/50
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0196 - val_loss:
0.0287
Epoch 22/50
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0161 - val_loss:
0.0289
Epoch 23/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0155 - val_loss:
0.0320
Epoch 24/50
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0168 - val_loss:
0.0232
```

```
Epoch 25/50
36/36 ━━━━━━━━━━ 1s 11ms/step - loss: 0.0139 - val_loss:
0.0274
Epoch 26/50
36/36 ━━━━━━━━ 0s 12ms/step - loss: 0.0132 - val_loss:
0.0259
Epoch 27/50
36/36 ━━━━━━ 0s 13ms/step - loss: 0.0118 - val_loss:
0.0282
Epoch 28/50
36/36 ━━━━ 1s 13ms/step - loss: 0.0113 - val_loss:
0.0225
Epoch 29/50
36/36 ━━━━ 0s 11ms/step - loss: 0.0110 - val_loss:
0.0267
Epoch 30/50
36/36 ━━━━ 1s 12ms/step - loss: 0.0102 - val_loss:
0.0206
Epoch 31/50
36/36 ━━━━ 1s 14ms/step - loss: 0.0104 - val_loss:
0.0264
Epoch 32/50
36/36 ━━━━ 0s 12ms/step - loss: 0.0116 - val_loss:
0.0197
Epoch 33/50
36/36 ━━━━ 0s 11ms/step - loss: 0.0108 - val_loss:
0.0198
Epoch 34/50
36/36 ━━━━ 0s 10ms/step - loss: 0.0103 - val_loss:
0.0178
Epoch 35/50
36/36 ━━━━ 0s 11ms/step - loss: 0.0095 - val_loss:
0.0200
Epoch 36/50
36/36 ━━━━ 0s 8ms/step - loss: 0.0099 - val_loss:
0.0175
Epoch 37/50
36/36 ━━━━ 0s 9ms/step - loss: 0.0108 - val_loss:
0.0187
Epoch 38/50
36/36 ━━━━ 0s 11ms/step - loss: 0.0094 - val_loss:
0.0192
Epoch 39/50
36/36 ━━━━ 0s 9ms/step - loss: 0.0105 - val_loss:
0.0298
Epoch 40/50
36/36 ━━━━ 0s 11ms/step - loss: 0.0110 - val_loss:
0.0158
Epoch 41/50
```

```
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0103 - val_loss:  
0.0252  
Epoch 42/50  
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0099 - val_loss:  
0.0176  
Epoch 43/50  
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0105 - val_loss:  
0.0243  
Epoch 44/50  
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0102 - val_loss:  
0.0226  
Epoch 45/50  
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0102 - val_loss:  
0.0241  
Epoch 46/50  
36/36 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0103 - val_loss:  
0.0176  
Epoch 47/50  
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0100 - val_loss:  
0.0214  
Epoch 48/50  
36/36 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0106 - val_loss:  
0.0202  
Epoch 49/50  
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0094 - val_loss:  
0.0195  
Epoch 50/50  
36/36 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0090 - val_loss:  
0.0240  
4/4 ━━━━━━━━━━ 0s 63ms/step
```



```
# Assuming you have fetched release data into a DataFrame
df_releases = pd.read_csv("github_all_data.csv")

# Convert the 'published_at' field to datetime
df_releases['published_at'] =
pd.to_datetime(df_releases['published_at'])

# Filter data between Nov 2022 and Nov 2024 (adjust as needed)
df_releases = df_releases[(df_releases['published_at'] >= start_date) & (df_releases['published_at'] <= end_date)]

# Group by the 'published_at' date and count the number of releases
df_releases_daily =
df_releases.groupby(df_releases['published_at'].dt.date).size().reset_index(name='release_count')

# Plot the release data over time
plt.figure(figsize=(10, 6))
plt.plot(df_releases_daily['published_at'],
df_releases_daily['release_count'], label='Release Count')
plt.xlabel('Date')
plt.ylabel('Number of Releases')
plt.title('Releases Over Time (Nov 2022 - Nov 2024)')
```

```

plt.xticks(rotation=45)
plt.legend()
plt.show()

# Prepare the time series data for LSTM
releases_values = df_releases_daily['release_count'].values.reshape(-1, 1)

# Normalize the data
scaler_releases = MinMaxScaler(feature_range=(0, 1))
releases_scaled = scaler_releases.fit_transform(releases_values)

# Split the data into training and testing sets (80% train, 20% test)
train_size = int(len(releases_scaled) * 0.8)
train_data, test_data = releases_scaled[:train_size],
releases_scaled[train_size:]

# Create the dataset for LSTM (X, y)
def create_dataset(data, time_step=1):
    X, y = [], []
    for i in range(len(data) - time_step):
        X.append(data[i:(i + time_step), 0])
        y.append(data[i + time_step, 0])
    return np.array(X), np.array(y)

# Reshape data into the form that can be used by LSTM (samples, time steps, features)
time_step = 30 # Use the past 30 days to predict the next day's releases
X_train, y_train = create_dataset(train_data, time_step)
X_test, y_test = create_dataset(test_data, time_step)

X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)

# Build the LSTM model
model_releases = Sequential()
model_releases.add(LSTM(units=50, return_sequences=False,
input_shape=(X_train.shape[1], 1)))
model_releases.add(Dense(units=1))

# Compile the model
model_releases.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
model_releases.fit(X_train, y_train, epochs=50, batch_size=16,
validation_data=(X_test, y_test), verbose=1)

# Predict the releases on the test set
predicted_releases = model_releases.predict(X_test)

```

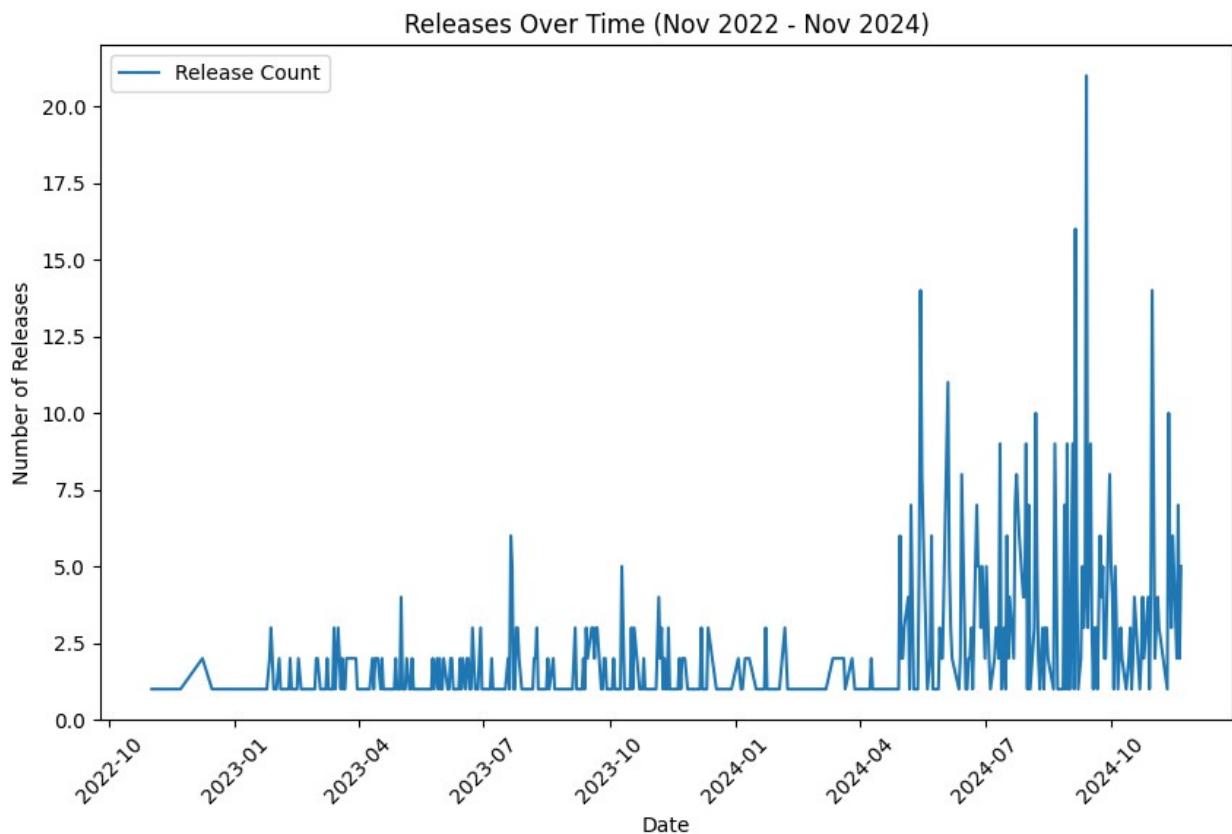
```

# Inverse transform the predicted values to get them back to original scale
predicted_releases =
scaler_releases.inverse_transform(predicted_releases)

# Inverse transform the actual values to get them back to original scale
y_test_actual = scaler_releases.inverse_transform(y_test.reshape(-1,
1))

# Plot the actual vs predicted releases
plt.figure(figsize=(10, 6))
plt.plot(df_releases_daily['published_at'][len(y_test_actual):],
y_test_actual, label='Actual Releases')
plt.plot(df_releases_daily['published_at'][len(predicted_releases):],
predicted_releases, label='Predicted Releases', linestyle='--')
plt.xlabel('Date')
plt.ylabel('Number of Releases')
plt.title('Releases: Actual vs Predicted (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.savefig("forecasting/Tensorflow_LSTM/releases_forecast.png")
plt.show()

```



```
Epoch 1/50
22/22 ━━━━━━━━━━ 2s 22ms/step - loss: 0.0052 - val_loss:
0.0417
Epoch 2/50
22/22 ━━━━━━━━ 0s 9ms/step - loss: 0.0081 - val_loss:
0.0398
Epoch 3/50
22/22 ━━━━━━ 0s 11ms/step - loss: 0.0049 - val_loss:
0.0393
Epoch 4/50
22/22 ━━━━ 0s 10ms/step - loss: 0.0063 - val_loss:
0.0386
Epoch 5/50
22/22 ━━ 0s 9ms/step - loss: 0.0042 - val_loss:
0.0387
Epoch 6/50
22/22 ━ 0s 10ms/step - loss: 0.0073 - val_loss:
0.0392
Epoch 7/50
22/22 0s 10ms/step - loss: 0.0056 - val_loss:
0.0386
Epoch 8/50
22/22 0s 10ms/step - loss: 0.0050 - val_loss:
0.0386
Epoch 9/50
22/22 0s 8ms/step - loss: 0.0048 - val_loss:
0.0384
Epoch 10/50
22/22 0s 10ms/step - loss: 0.0043 - val_loss:
0.0381
Epoch 11/50
22/22 0s 10ms/step - loss: 0.0039 - val_loss:
0.0378
Epoch 12/50
22/22 0s 11ms/step - loss: 0.0049 - val_loss:
0.0387
Epoch 13/50
22/22 0s 12ms/step - loss: 0.0038 - val_loss:
0.0379
Epoch 14/50
22/22 0s 10ms/step - loss: 0.0055 - val_loss:
0.0385
Epoch 15/50
22/22 0s 10ms/step - loss: 0.0070 - val_loss:
0.0394
Epoch 16/50
22/22 0s 10ms/step - loss: 0.0043 - val_loss:
0.0387
Epoch 17/50
22/22 0s 8ms/step - loss: 0.0063 - val_loss:
```

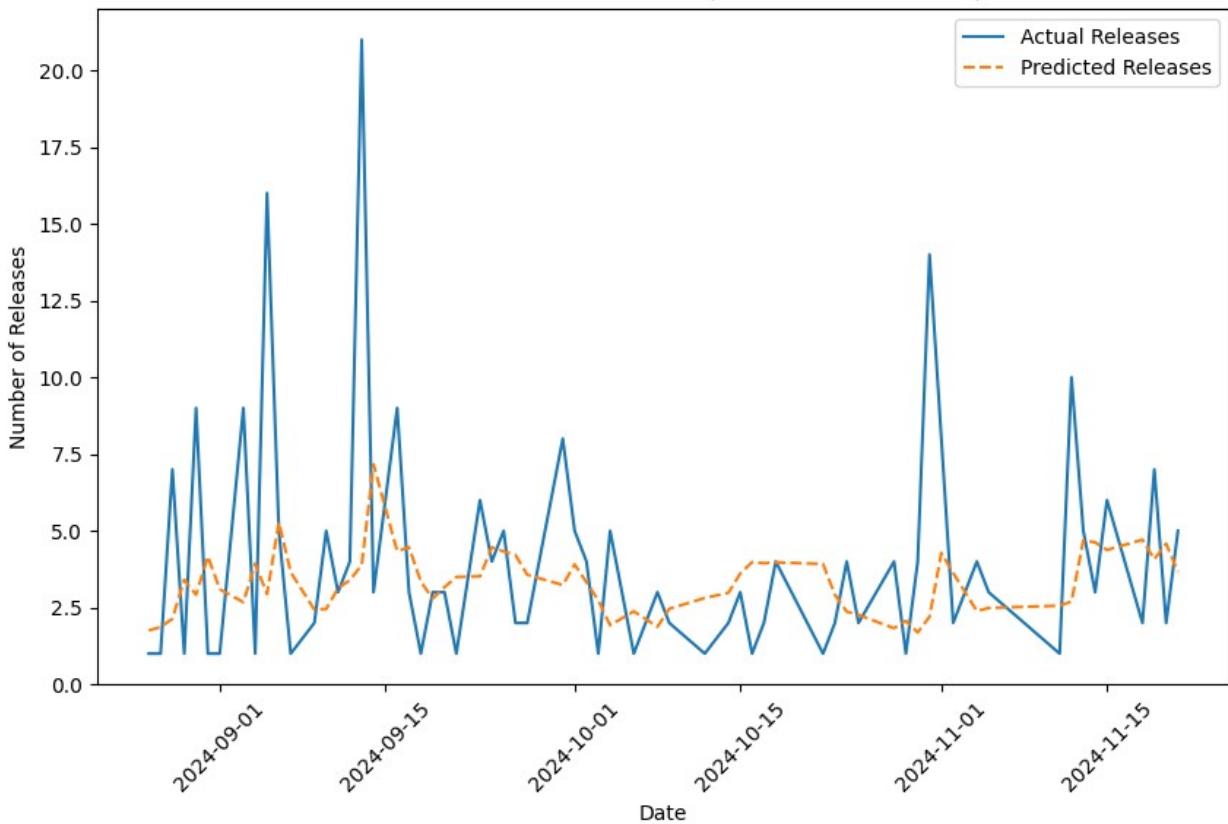
```
0.0394
Epoch 18/50
22/22 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0037 - val_loss:
0.0385
Epoch 19/50
22/22 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0058 - val_loss:
0.0389
Epoch 20/50
22/22 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0043 - val_loss:
0.0383
Epoch 21/50
22/22 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0047 - val_loss:
0.0393
Epoch 22/50
22/22 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0044 - val_loss:
0.0385
Epoch 23/50
22/22 ━━━━━━━━━━ 0s 9ms/step - loss: 0.0030 - val_loss:
0.0380
Epoch 24/50
22/22 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0050 - val_loss:
0.0386
Epoch 25/50
22/22 ━━━━━━━━━━ 0s 13ms/step - loss: 0.0054 - val_loss:
0.0385
Epoch 26/50
22/22 ━━━━━━━━━━ 0s 16ms/step - loss: 0.0047 - val_loss:
0.0384
Epoch 27/50
22/22 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0033 - val_loss:
0.0379
Epoch 28/50
22/22 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0047 - val_loss:
0.0383
Epoch 29/50
22/22 ━━━━━━━━━━ 0s 13ms/step - loss: 0.0035 - val_loss:
0.0385
Epoch 30/50
22/22 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0056 - val_loss:
0.0386
Epoch 31/50
22/22 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0031 - val_loss:
0.0387
Epoch 32/50
22/22 ━━━━━━━━━━ 0s 12ms/step - loss: 0.0051 - val_loss:
0.0383
Epoch 33/50
22/22 ━━━━━━━━━━ 0s 10ms/step - loss: 0.0044 - val_loss:
0.0382
Epoch 34/50
```

```
22/22 ━━━━━━━━━━ 0s 11ms/step - loss: 0.0033 - val_loss:  
0.0384  
Epoch 35/50  
22/22 ━━━━━━━━ 0s 12ms/step - loss: 0.0052 - val_loss:  
0.0399  
Epoch 36/50  
22/22 ━━━━━━ 0s 11ms/step - loss: 0.0037 - val_loss:  
0.0387  
Epoch 37/50  
22/22 ━━━━ 0s 14ms/step - loss: 0.0043 - val_loss:  
0.0384  
Epoch 38/50  
22/22 ━━ 0s 11ms/step - loss: 0.0050 - val_loss:  
0.0381  
Epoch 39/50  
22/22 ━ 0s 12ms/step - loss: 0.0039 - val_loss:  
0.0382  
Epoch 40/50  
22/22 0s 10ms/step - loss: 0.0038 - val_loss:  
0.0384  
Epoch 41/50  
22/22 0s 10ms/step - loss: 0.0039 - val_loss:  
0.0387  
Epoch 42/50  
22/22 0s 12ms/step - loss: 0.0037 - val_loss:  
0.0382  
Epoch 43/50  
22/22 0s 11ms/step - loss: 0.0052 - val_loss:  
0.0386  
Epoch 44/50  
22/22 0s 9ms/step - loss: 0.0047 - val_loss:  
0.0386  
Epoch 45/50  
22/22 0s 8ms/step - loss: 0.0044 - val_loss:  
0.0387  
Epoch 46/50  
22/22 0s 15ms/step - loss: 0.0047 - val_loss:  
0.0384  
Epoch 47/50  
22/22 0s 10ms/step - loss: 0.0040 - val_loss:  
0.0385  
Epoch 48/50  
22/22 0s 10ms/step - loss: 0.0035 - val_loss:  
0.0392  
Epoch 49/50  
22/22 0s 9ms/step - loss: 0.0045 - val_loss:  
0.0388  
Epoch 50/50  
22/22 0s 11ms/step - loss: 0.0051 - val_loss:
```

0.0391

2/2 ————— 0s 18ms/step

Releases: Actual vs Predicted (Nov 2022 - Nov 2024)



```
import pandas as pd
import matplotlib.pyplot as plt
from prophet import Prophet

# Load your dataset
df = pd.read_csv("github_all_data.csv")

# Ensure 'created_at' is converted to datetime
df['created_at'] = pd.to_datetime(df['created_at'], errors='coerce')

# Check if the datetime is timezone-aware, and localize to UTC if not
if df['created_at'].dt.tz is None:
    df['created_at'] = df['created_at'].dt.tz_localize('UTC')
else:
    df['created_at'] = df['created_at'].dt.tz_convert('UTC')

# Remove timezone information from 'created_at' before using Prophet
df['created_at'] = df['created_at'].dt.tz_localize(None)

# Filter data for the past 2 years
```

```

end_date = pd.to_datetime('2024-11-21')
start_date = end_date - pd.DateOffset(years=2)

# Filter the dataframe for the relevant date range
df = df[(df['created_at'] >= start_date) & (df['created_at'] <= end_date)]

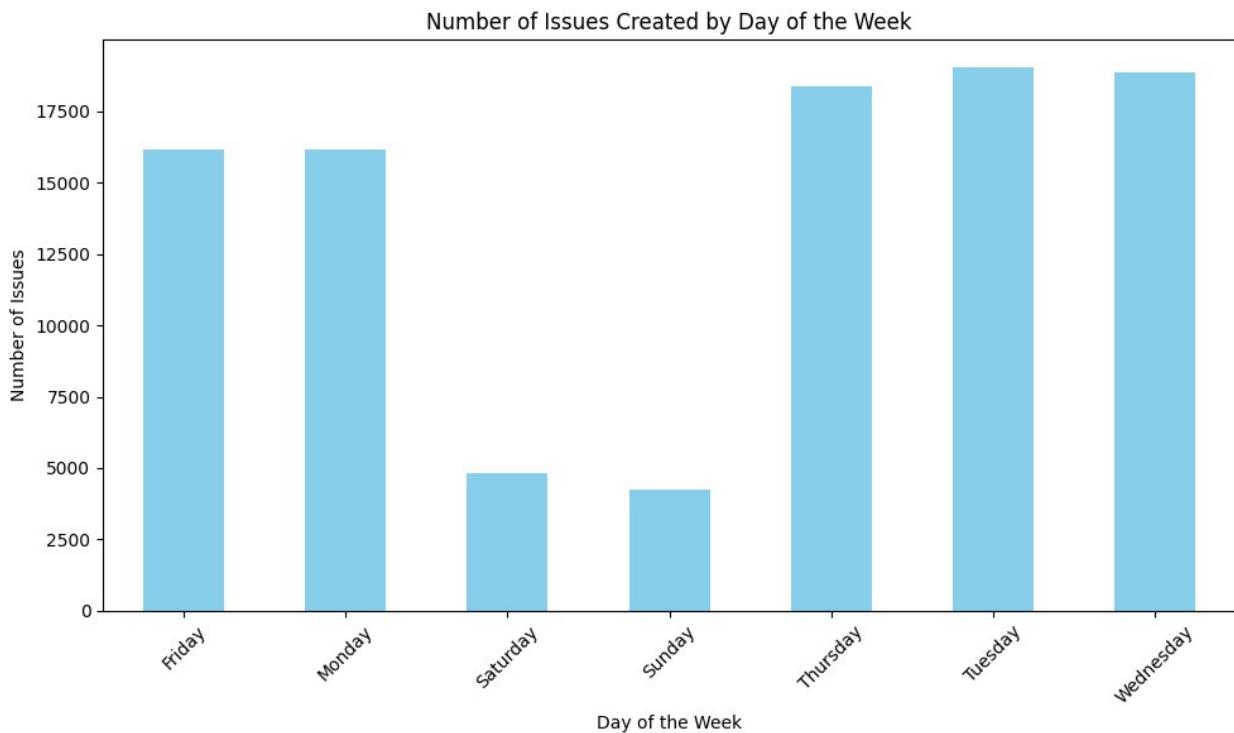
# Drop any rows with NaN values in relevant columns
df = df.dropna(subset=['created_at', 'repository', 'type'])

# Get the day of the week when issues were created
df['day_of_week'] = df['created_at'].dt.day_name()

# Count issues created each day of the week
day_of_week_counts = df['day_of_week'].value_counts().sort_index()

# Plot the day of the week with the most issues created
plt.figure(figsize=(10, 6))
day_of_week_counts.plot(kind='bar', color='skyblue')
plt.title("Number of Issues Created by Day of the Week")
plt.xlabel("Day of the Week")
plt.ylabel("Number of Issues")
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("forecasting/Prophet/number_of_issues_created.png")
plt.show()

```



```

#Add your code for requirement 9.2 in this cell

# Ensure 'closed_at' is converted to datetime
df['closed_at'] = pd.to_datetime(df['closed_at'], errors='coerce')

# Check if the datetime is timezone-aware, and localize to UTC if not
if df['closed_at'].dt.tz is None:
    df['closed_at'] = df['closed_at'].dt.tz_localize('UTC')
else:
    df['closed_at'] = df['closed_at'].dt.tz_convert('UTC')

# Remove timezone information from 'closed_at' before using Prophet
df['closed_at'] = df['closed_at'].dt.tz_localize(None)

# Filter data for the past 2 years
df = df[(df['closed_at'] >= start_date) & (df['closed_at'] <=
end_date)]

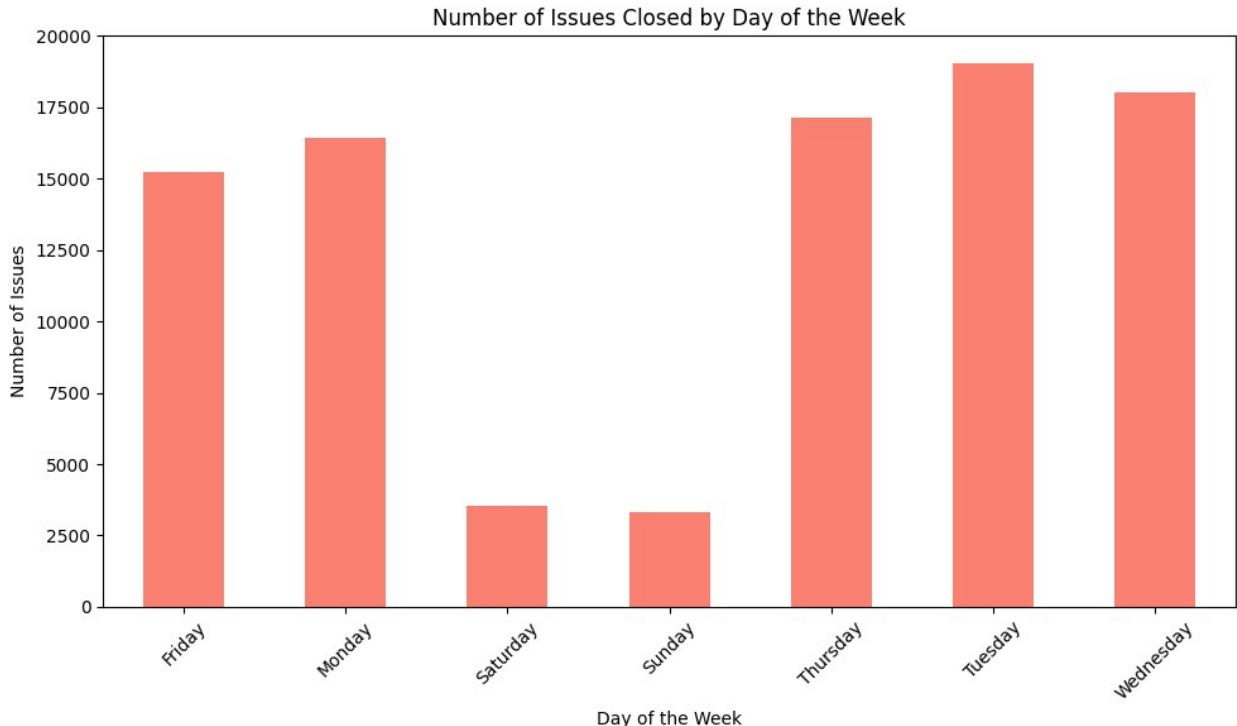
# Drop any rows with NaN values in relevant columns
df = df.dropna(subset=['closed_at', 'repository', 'type'])

# Get the day of the week when issues were closed
df['closed_day_of_week'] = df['closed_at'].dt.day_name()

# Count issues closed each day of the week
closed_day_of_week_counts =
df['closed_day_of_week'].value_counts().sort_index()

# Plot the day of the week with the most issues closed
plt.figure(figsize=(10, 6))
closed_day_of_week_counts.plot(kind='bar', color='salmon')
plt.title("Number of Issues Closed by Day of the Week")
plt.xlabel("Day of the Week")
plt.ylabel("Number of Issues")
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("forecasting/Prophet/number_of_issues_closed.png")
plt.show()

```



```
#Add your code for requirement 9.3 in this cell
# Ensure 'closed_at' is converted to datetime
df['closed_at'] = pd.to_datetime(df['closed_at'], errors='coerce')

# Check if the datetime is timezone-aware, and localize to UTC if not
if df['closed_at'].dt.tz is None:
    df['closed_at'] = df['closed_at'].dt.tz_localize('UTC')
else:
    df['closed_at'] = df['closed_at'].dt.tz_convert('UTC')

# Remove timezone information from 'closed_at' before using Prophet
df['closed_at'] = df['closed_at'].dt.tz_localize(None)

# Filter data for the past 2 years
df = df[(df['closed_at'] >= start_date) & (df['closed_at'] <=
end_date)]

# Drop any rows with NaN values in relevant columns
df = df.dropna(subset=['closed_at', 'repository', 'type'])

# Get the month of the year when issues were closed
df['closed_month'] = df['closed_at'].dt.month_name()

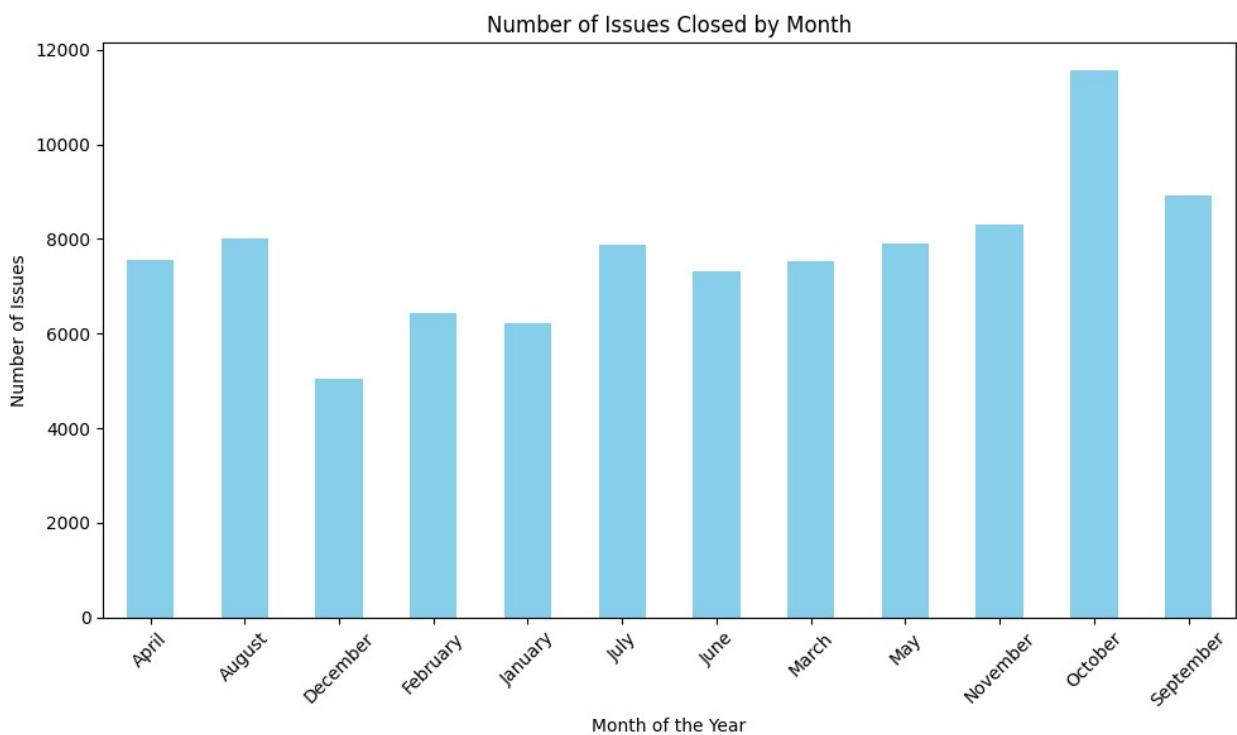
# Count issues closed each month
closed_month_counts = df['closed_month'].value_counts().sort_index()

# Plot the month with the most issues closed
```

```

plt.figure(figsize=(10, 6))
closed_month_counts.plot(kind='bar', color='skyblue')
plt.title("Number of Issues Closed by Month")
plt.xlabel("Month of the Year")
plt.ylabel("Number of Issues")
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("forecasting/Prophet/max_number_of_issues_closed_by_month.
png")
plt.show()

```



```

import pandas as pd
import matplotlib.pyplot as plt
from prophet import Prophet

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter the data for 'issue' type and 'created' status
df_created_issues = df_detailed[df_detailed['type'] == 'issue']

# Convert 'created_at' to datetime
df_created_issues['created_at'] =
pd.to_datetime(df_created_issues['created_at'])

# Filter data between November 2022 and November 2024
start_date = '2022-11-01'

```

```

end_date = '2024-11-30'
df_created_issues = df_created_issues[(df_created_issues['created_at'] >= start_date) & (df_created_issues['created_at'] <= end_date)]

# Extract the date part (ignoring time) and group by date to count created issues
df_created_issues['date'] = df_created_issues['created_at'].dt.date
df_created_daily =
df_created_issues.groupby('date').size().reset_index(name='created_issue_count')

# Prepare the data for Prophet (Prophet expects columns 'ds' for date and 'y' for the values to forecast)
prophet_df = df_created_daily.rename(columns={'date': 'ds',
'created_issue_count': 'y'})

# Initialize and fit the Prophet model
model = Prophet(daily_seasonality=True)
model.fit(prophet_df)

# Forecast the data based on existing dates
forecast = model.predict(prophet_df)

# Plot the created issues forecast
plt.figure(figsize=(10, 6))
model.plot(forecast)
plt.title('Created Issues Forecast (Nov 2022 - Nov 2024)')
plt.xlabel('Date')
plt.ylabel('Number of Created Issues')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("forecasting/Prophet/created_issues_forecast.png")
plt.show()

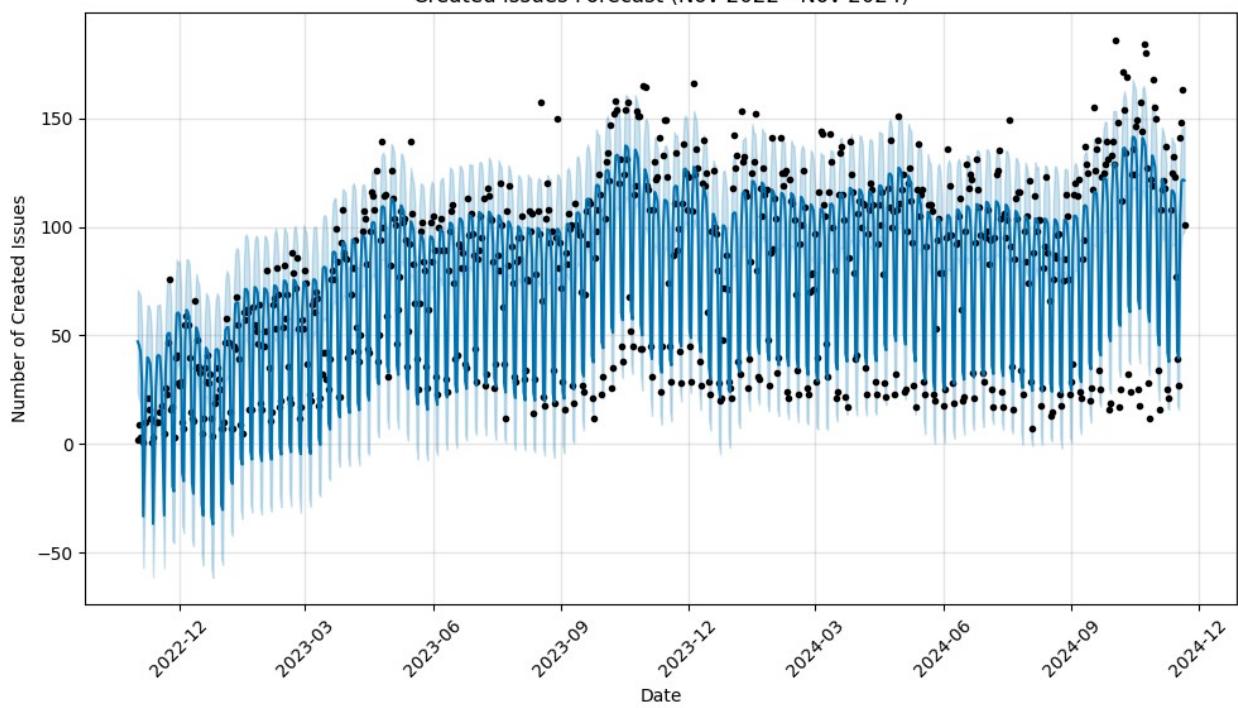
# Optional: Plot forecast components (trend, yearly, and weekly seasonality)
model.plot_components(forecast)
plt.show()

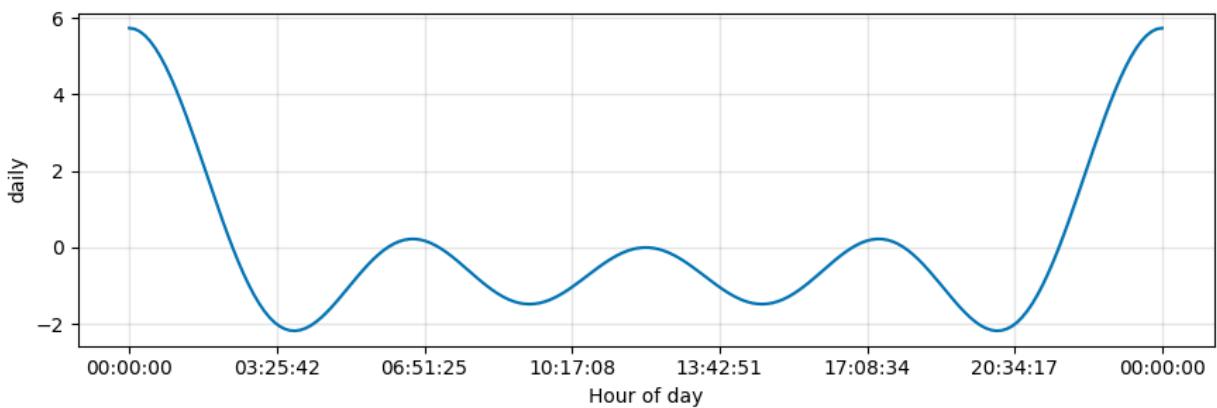
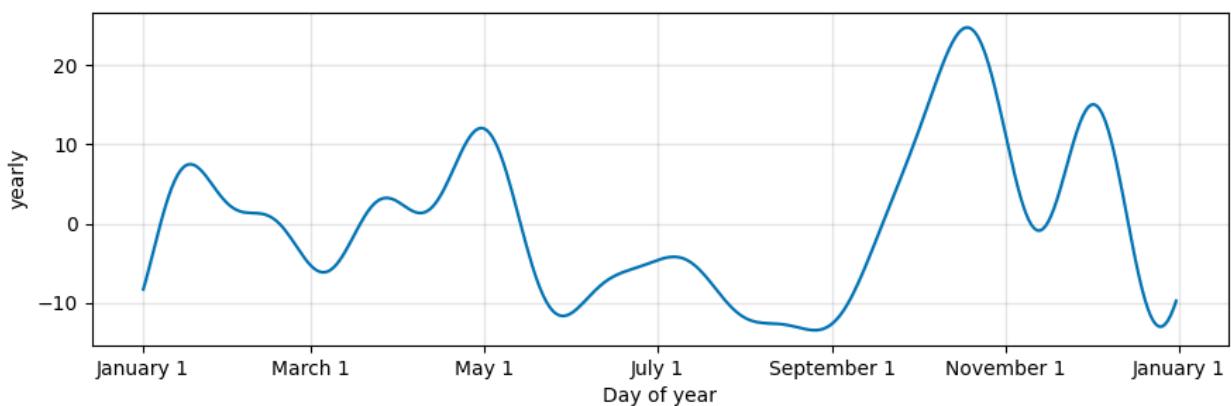
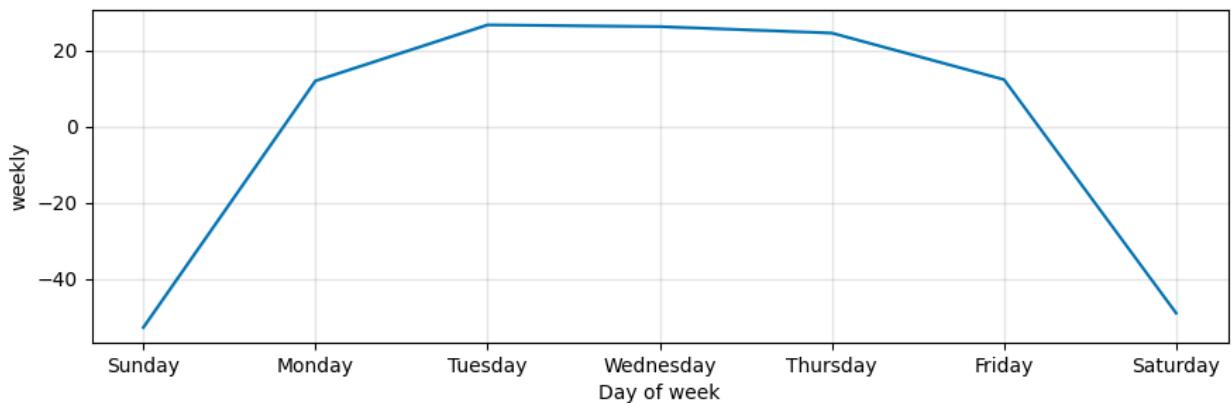
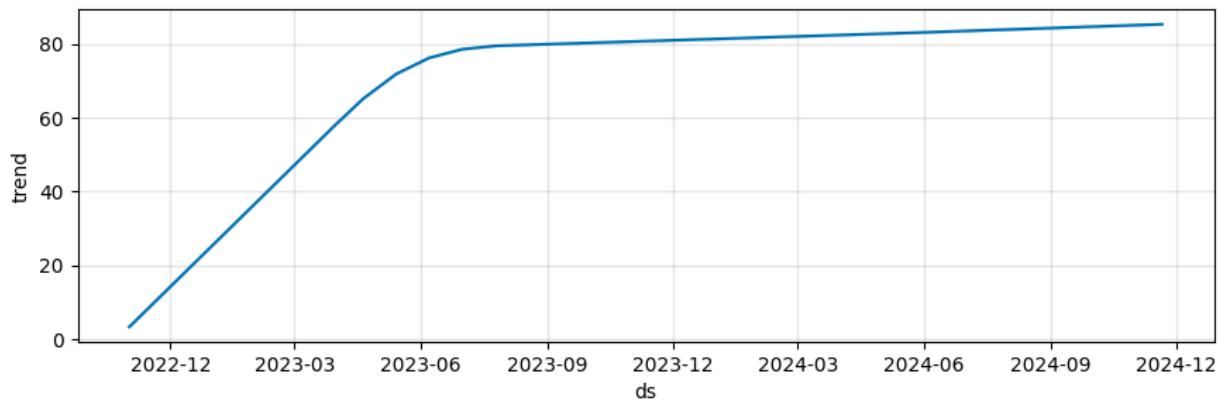
22:34:39 - cmdstanpy - INFO - Chain [1] start processing
22:34:40 - cmdstanpy - INFO - Chain [1] done processing

<Figure size 1000x600 with 0 Axes>

```

Created Issues Forecast (Nov 2022 - Nov 2024)





```

#Add your code for requirement 9.5 in this cell

import pandas as pd
import matplotlib.pyplot as plt
from prophet import Prophet

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter the data for 'issue' type and 'closed' status
df_closed_issues = df_detailed[df_detailed['type'] == 'issue']
df_closed_issues = df_closed_issues[df_closed_issues['state'] ==
'closed']

# Convert 'closed_at' to datetime
df_closed_issues['closed_at'] =
pd.to_datetime(df_closed_issues['closed_at'])

# Filter data between November 2022 and November 2024
start_date = '2022-11-01'
end_date = '2024-11-30'
df_closed_issues = df_closed_issues[(df_closed_issues['closed_at'] >=
start_date) & (df_closed_issues['closed_at'] <= end_date)]

# Extract the date part (ignoring time) and group by date to count
# closed issues
df_closed_issues['date'] = df_closed_issues['closed_at'].dt.date
df_closed_daily =
df_closed_issues.groupby('date').size().reset_index(name='closed_issue
_count')

# Prepare the data for Prophet (Prophet expects columns 'ds' for date
# and 'y' for the values to forecast)
prophet_df_closed = df_closed_daily.rename(columns={'date': 'ds',
'closed_issue_count': 'y'})

# Initialize and fit the Prophet model
model_closed_issues = Prophet(daily_seasonality=True)
model_closed_issues.fit(prophet_df_closed)

# Forecast the data based on existing dates
forecast_closed = model_closed_issues.predict(prophet_df_closed)

# Plot the closed issues forecast
plt.figure(figsize=(10, 6))
model_closed_issues.plot(forecast_closed)
plt.title('Closed Issues Forecast (Nov 2022 - Nov 2024)')
plt.xlabel('Date')
plt.ylabel('Number of Closed Issues')
plt.xticks(rotation=45)

```

```

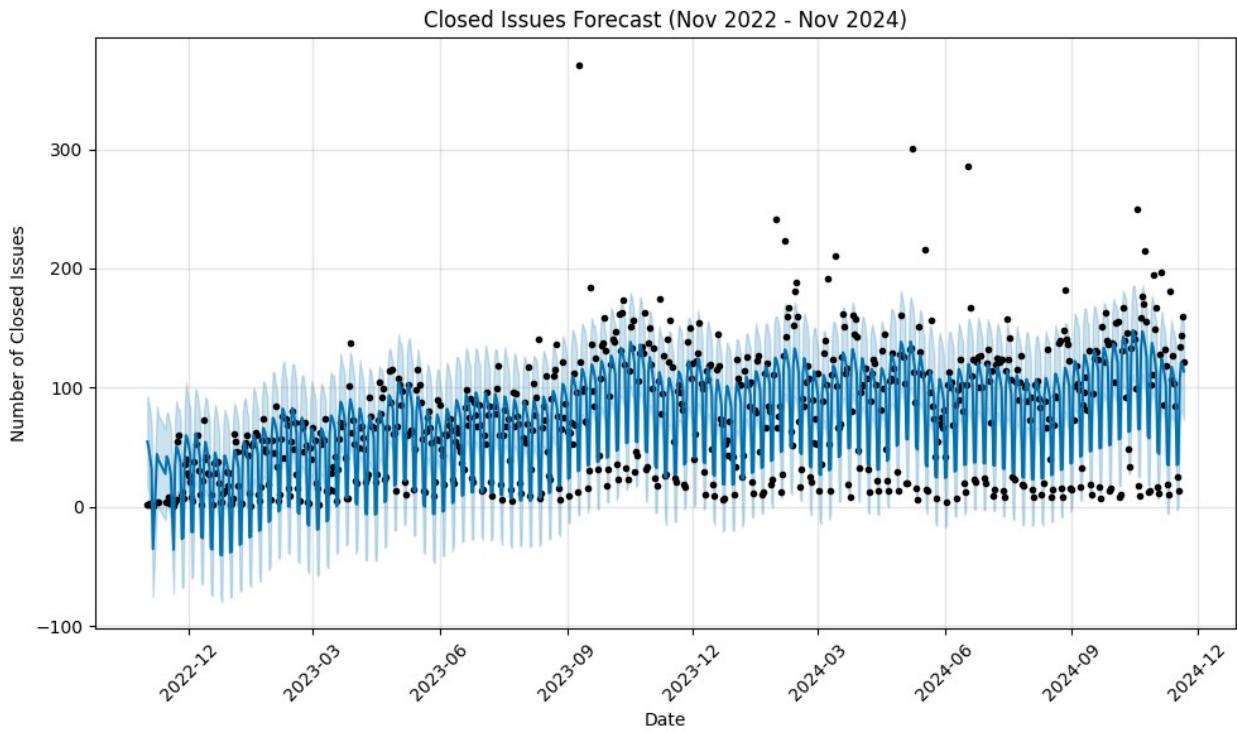
plt.tight_layout()
plt.savefig("forecasting/Prophet/closed_issues_forecast.png")
plt.show()

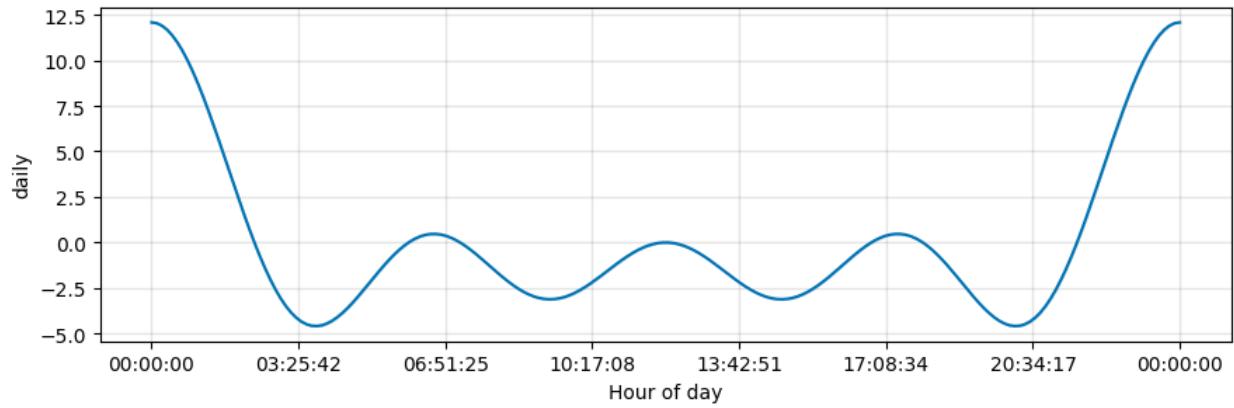
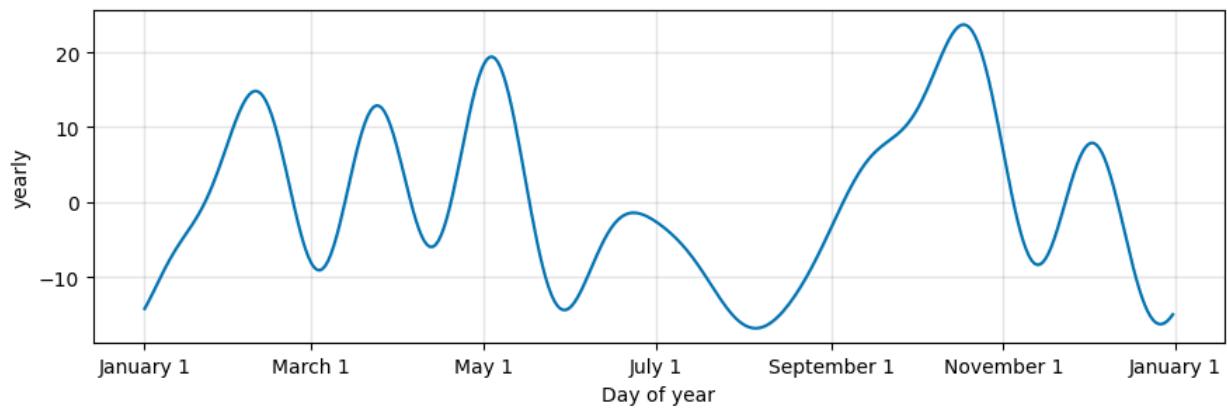
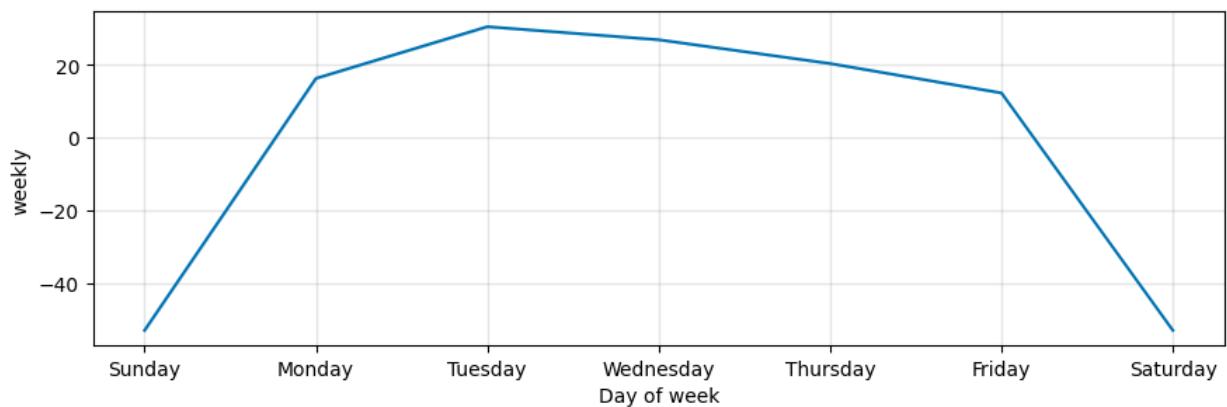
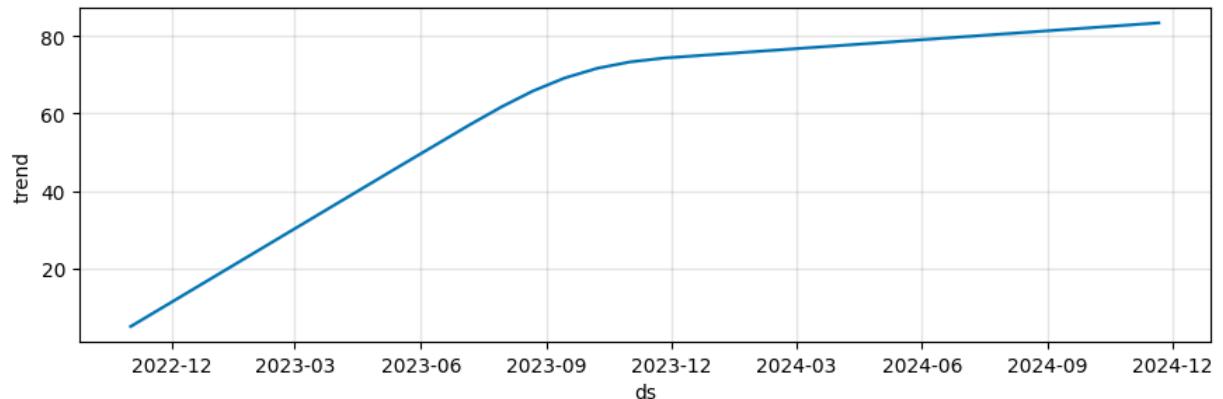
# Optional: Plot forecast components (trend, yearly, and weekly
seasonality)
model_closed_issues.plot_components(forecast_closed)
plt.show()

```

22:35:07 - cmdstanpy - INFO - Chain [1] start processing
22:35:07 - cmdstanpy - INFO - Chain [1] done processing

<Figure size 1000x600 with 0 Axes>





```

#Add your code for requirement 9.6 in this cell

import pandas as pd
import matplotlib.pyplot as plt
from prophet import Prophet

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter the data for 'pull_request' type
df_pulls = df_detailed[df_detailed['type'] == 'pull_request']

# Convert 'created_at' to datetime (assuming pulls have 'created_at' column)
df_pulls['created_at'] = pd.to_datetime(df_pulls['created_at'])

# Filter data between November 2022 and November 2024
start_date = '2022-11-01'
end_date = '2024-11-30'
df_pulls = df_pulls[(df_pulls['created_at'] >= start_date) &
(df_pulls['created_at'] <= end_date)]

# Extract the date part (ignoring time) and group by date to count pull requests
df_pulls['date'] = df_pulls['created_at'].dt.date
df_pulls_daily =
df_pulls.groupby('date').size().reset_index(name='pull_request_count')

# Prepare the data for Prophet (Prophet expects columns 'ds' for date and 'y' for the values to forecast)
prophet_df_pulls = df_pulls_daily.rename(columns={'date': 'ds',
'pull_request_count': 'y'})

# Initialize and fit the Prophet model
model_pulls = Prophet(daily_seasonality=True)
model_pulls.fit(prophet_df_pulls)

# Forecast the data based on existing dates
forecast_pulls = model_pulls.predict(prophet_df_pulls)

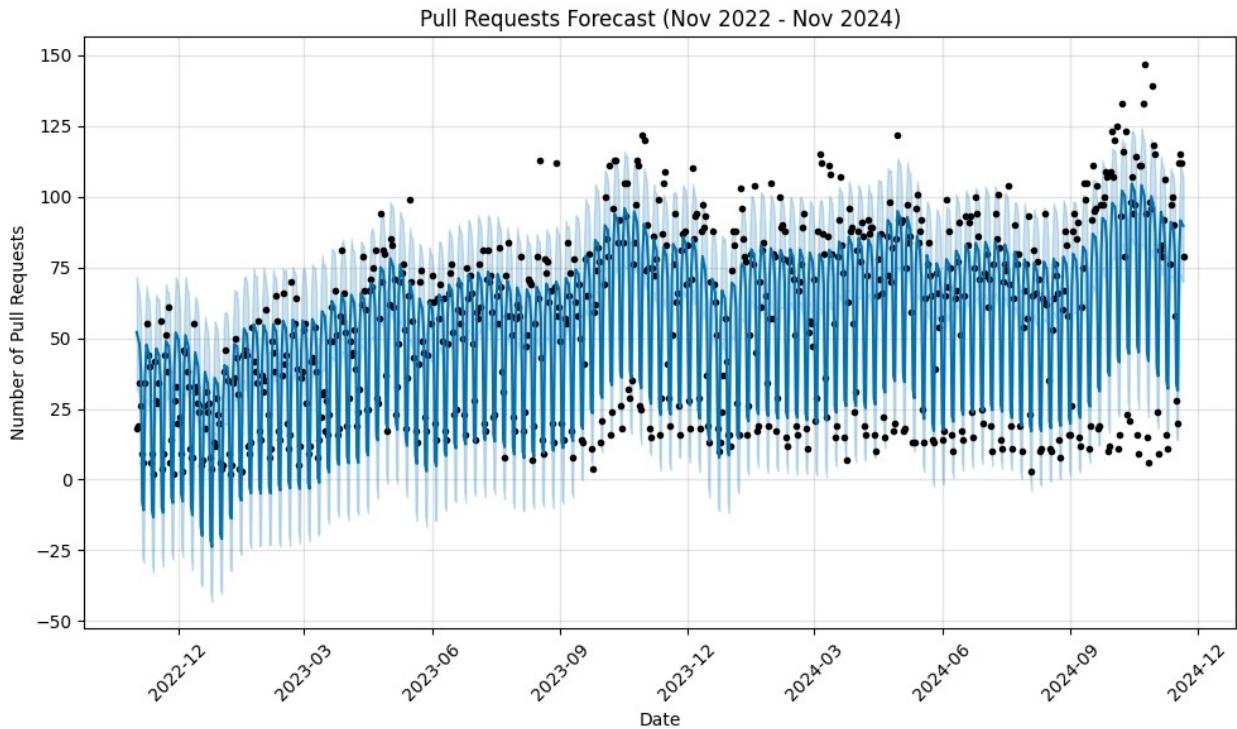
# Plot the pull requests forecast
plt.figure(figsize=(10, 6))
model_pulls.plot(forecast_pulls)
plt.title('Pull Requests Forecast (Nov 2022 - Nov 2024)')
plt.xlabel('Date')
plt.ylabel('Number of Pull Requests')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("forecasting/Prophet/pulls_forecast.png")
plt.show()

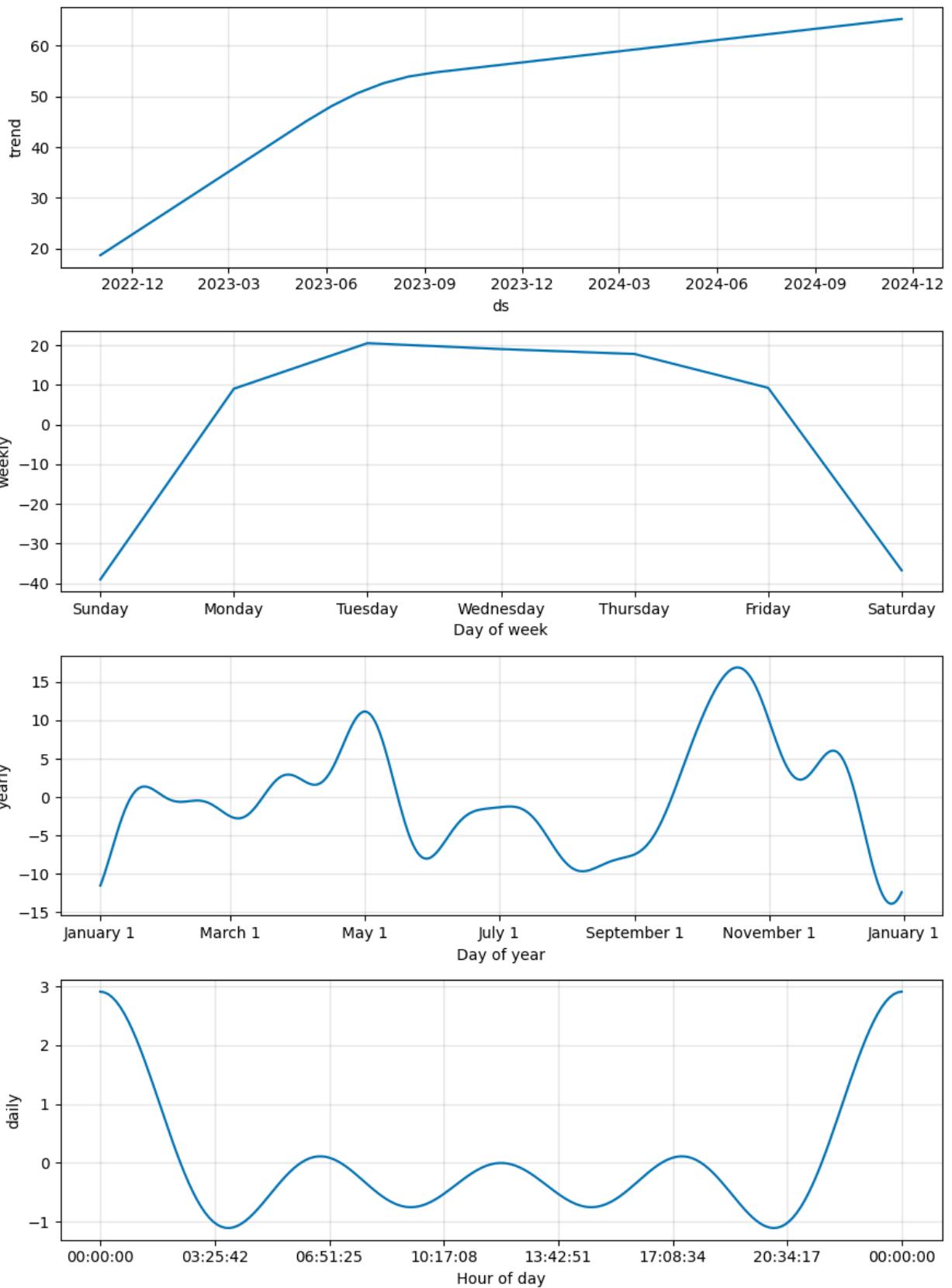
```

```
# Optional: Plot forecast components (trend, yearly, and weekly seasonality)
model_pulls.plot_components(forecast_pulls)
plt.show()
```

```
22:35:25 - cmdstanpy - INFO - Chain [1] start processing
22:35:25 - cmdstanpy - INFO - Chain [1] done processing
```

```
<Figure size 1000x600 with 0 Axes>
```





```

import pandas as pd
import matplotlib.pyplot as plt
from prophet import Prophet

# Load the data
df_commits = pd.read_csv("github_all_data.csv")

# Extract commit date (assuming the 'created_at' column contains
# commit timestamp)
df_commits['created_at'] = pd.to_datetime(df_commits['created_at'])

# Filter commits from November 2022 to November 2024 (adjust dates if
# needed)
start_date = '2022-11-01'
end_date = '2024-11-30'
df_commits = df_commits[(df_commits['created_at'] >= start_date) &
(df_commits['created_at'] <= end_date)]

# Group by date and count the number of commits per day
df_commits['date'] = df_commits['created_at'].dt.date
df_commits_daily =
df_commits.groupby('date').size().reset_index(name='commit_count')

# Prepare the data for Prophet (Prophet expects columns 'ds' for date
# and 'y' for the values to forecast)
prophet_df_commits = df_commits_daily.rename(columns={'date': 'ds',
'commit_count': 'y'})

# Initialize the Prophet model
model_commits = Prophet(daily_seasonality=True)

# Fit the model with the historical data
model_commits.fit(prophet_df_commits)

# Create a custom date range for the future predictions (next 2 years)
# For example, from the last date in the dataset, generate a list of
# future dates
last_date = prophet_df_commits['ds'].max()
future_dates = pd.date_range(start=last_date, periods=365*2 + 1,
freq='D')[1:]

# Prepare the dataframe for future dates
future_df = pd.DataFrame(future_dates, columns=['ds'])

# Forecast the future values
forecast_commits = model_commits.predict(future_df)

# Plot the commits forecast
plt.figure(figsize=(10, 6))
model_commits.plot(forecast_commits)
plt.title('Commits Forecast (Nov 2022 - Nov 2024)')

```

```

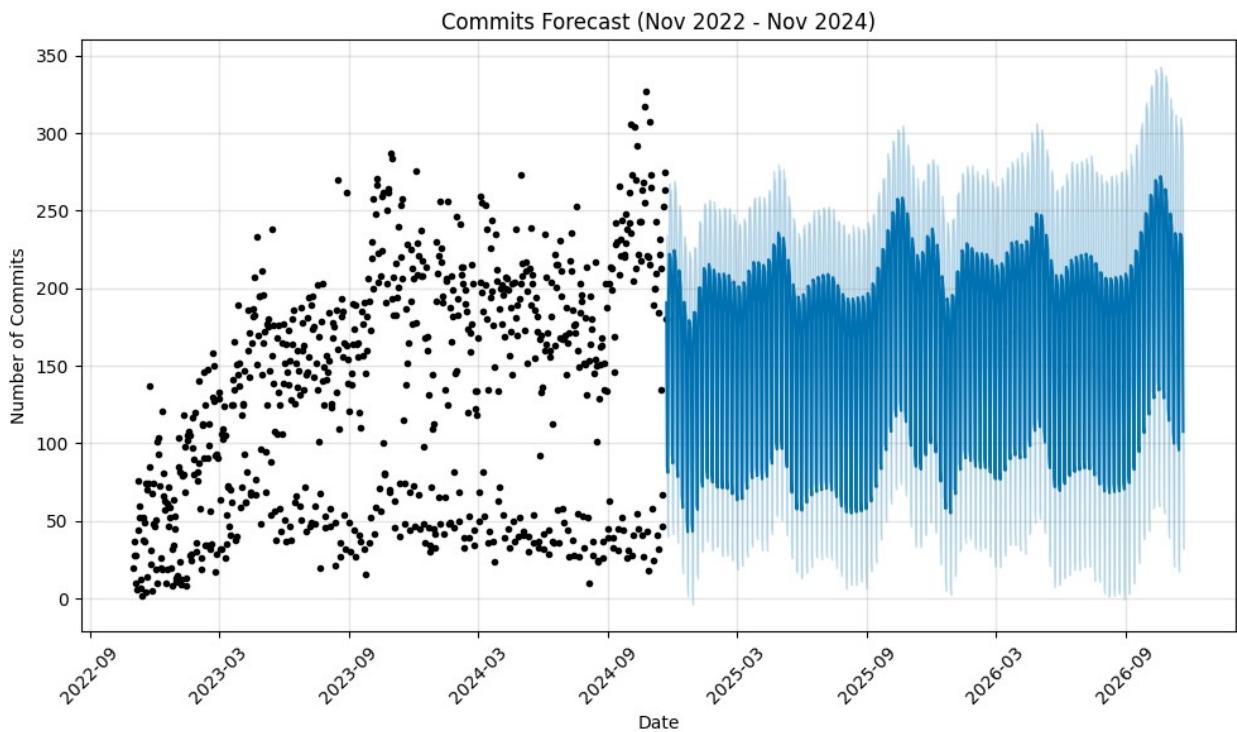
plt.xlabel('Date')
plt.ylabel('Number of Commits')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("forecasting/Prophet/commits_forecast.png")
plt.show()

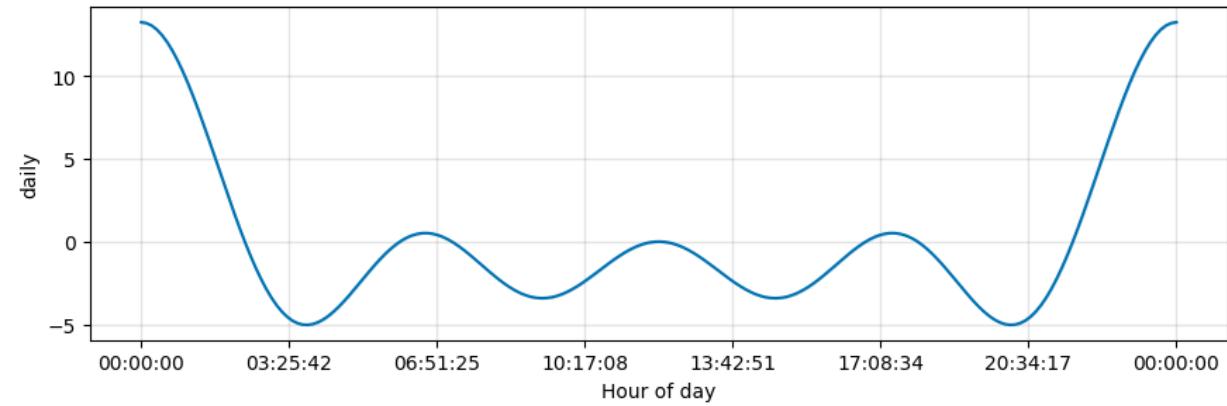
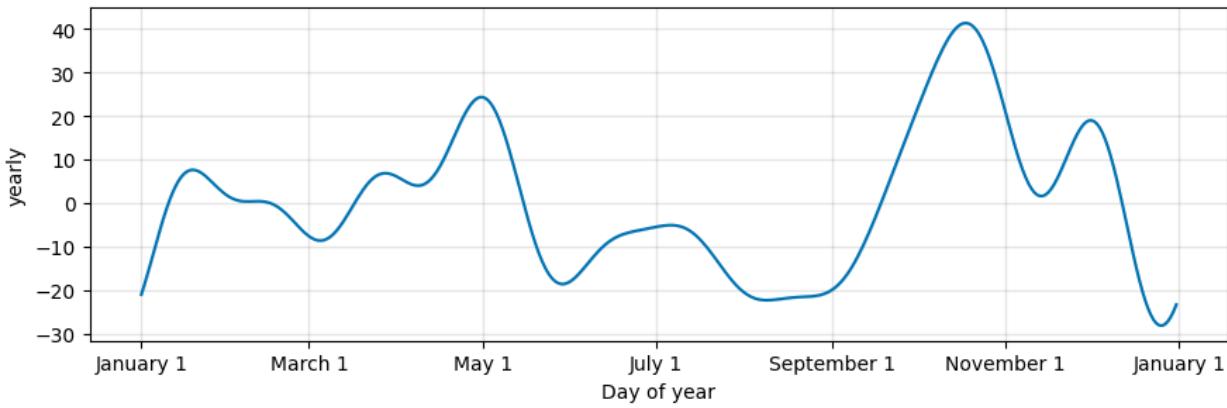
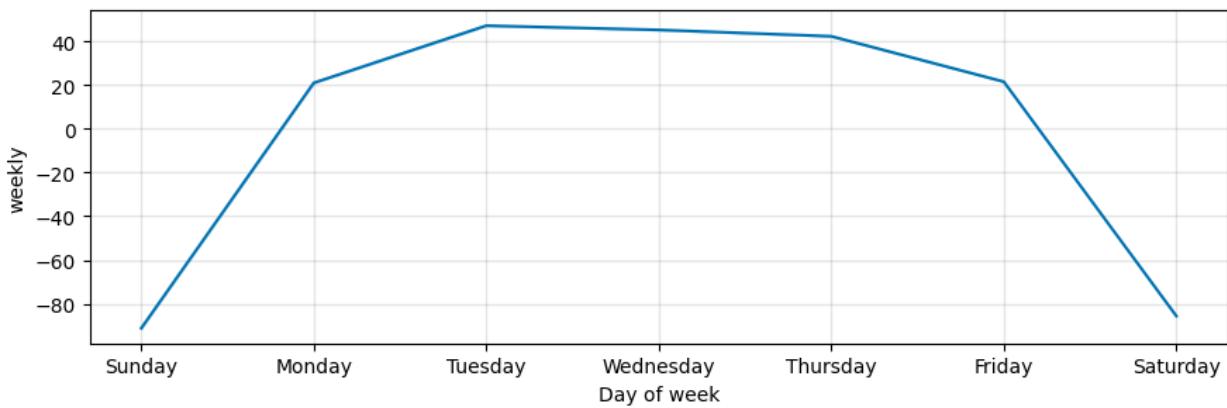
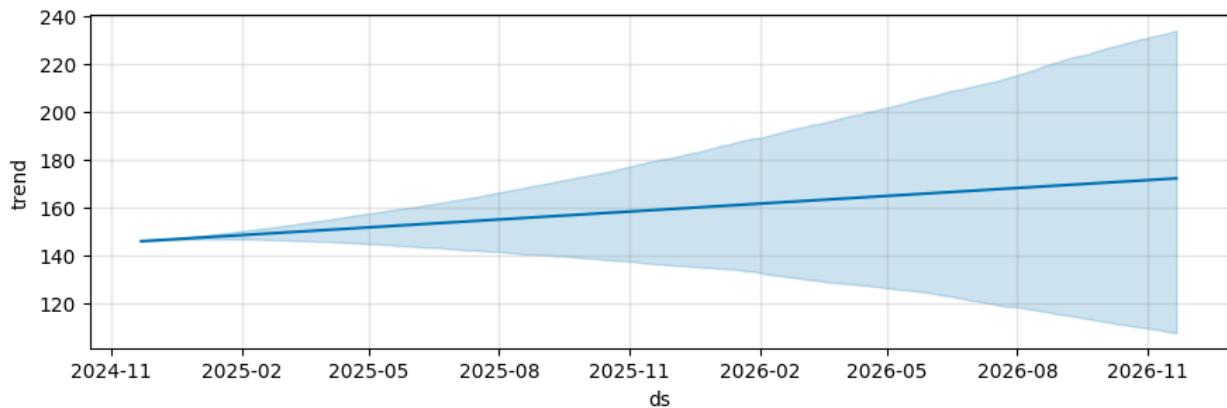
# Optional: Plot forecast components (trend, yearly, and weekly seasonality)
model_commits.plot_components(forecast_commits)
plt.show()

22:35:44 - cmdstanpy - INFO - Chain [1] start processing
22:35:44 - cmdstanpy - INFO - Chain [1] done processing

<Figure size 1000x600 with 0 Axes>

```





```

import pandas as pd
import matplotlib.pyplot as plt
from prophet import Prophet

# Load the data
df_branches = pd.read_csv("github_all_data.csv")

# Assuming there is a 'branch' or 'ref' column to track branch names
# Extract the date (assuming 'created_at' column contains timestamp)
df_branches['created_at'] = pd.to_datetime(df_branches['created_at'])

# Filter data between Nov 2022 and Nov 2024 (adjust as necessary)
start_date = '2022-11-01'
end_date = '2024-11-30'
df_branches = df_branches[(df_branches['created_at'] >= start_date) &
                           (df_branches['created_at'] <= end_date)]

# Filter for the 'branch' column (assuming it exists)
# Group by date and count the number of branches per day
df_branches['date'] = df_branches['created_at'].dt.date
df_branches_daily =
df_branches.groupby('date').size().reset_index(name='branch_count')

# Prepare the data for Prophet (Prophet expects columns 'ds' for date
# and 'y' for the values to forecast)
prophet_df_branches = df_branches_daily.rename(columns={'date': 'ds',
                                                         'branch_count': 'y'})

# Initialize the Prophet model
model_branches = Prophet(daily_seasonality=True)

# Fit the model with the historical data
model_branches.fit(prophet_df_branches)

# Create a custom date range for the future predictions (next 2 years)
# For example, from the last date in the dataset, generate a list of
# future dates
last_date = prophet_df_branches['ds'].max()
future_dates = pd.date_range(start=last_date, periods=365*2 + 1,
                             freq='D')[1:]

# Prepare the dataframe for future dates
future_df = pd.DataFrame(future_dates, columns=['ds'])

# Forecast the future values
forecast_branches = model_branches.predict(future_df)

# Plot the branches forecast
plt.figure(figsize=(10, 6))
model_branches.plot(forecast_branches)
plt.title('Branches Forecast (Nov 2022 - Nov 2024)')

```

```

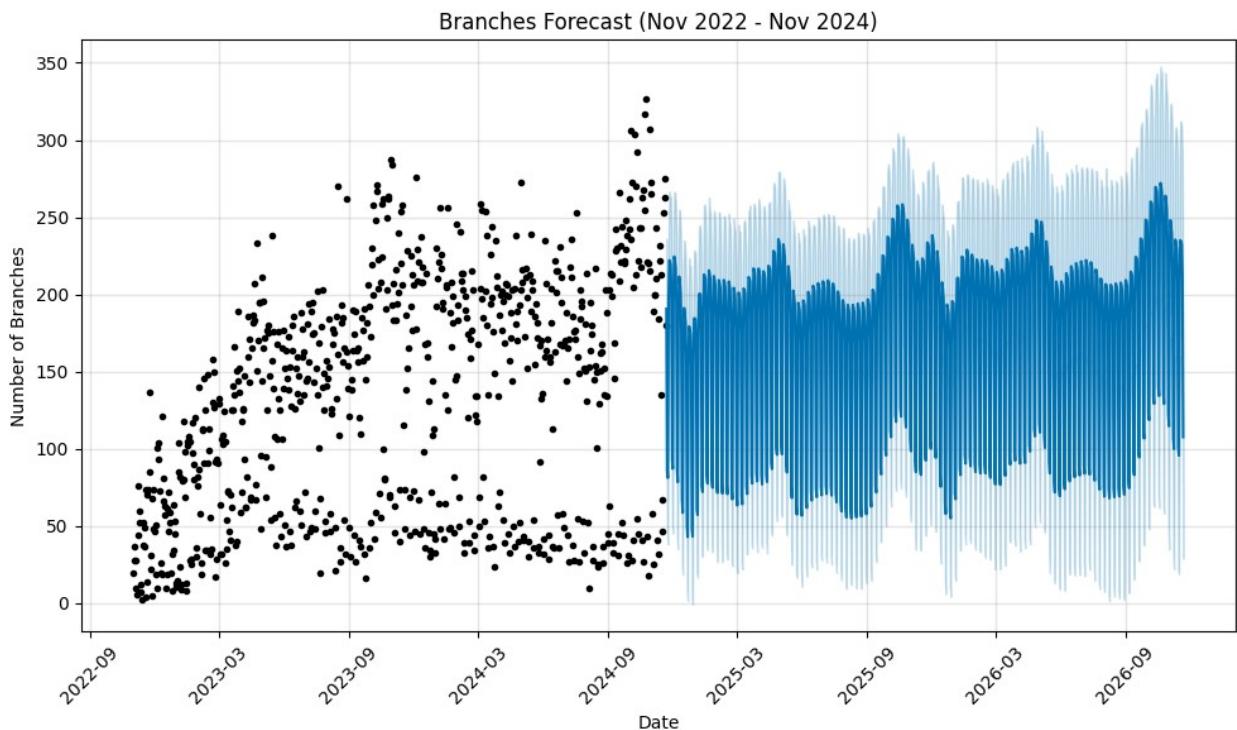
plt.xlabel('Date')
plt.ylabel('Number of Branches')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("forecasting/Prophet/branches_forecast.png")
plt.show()

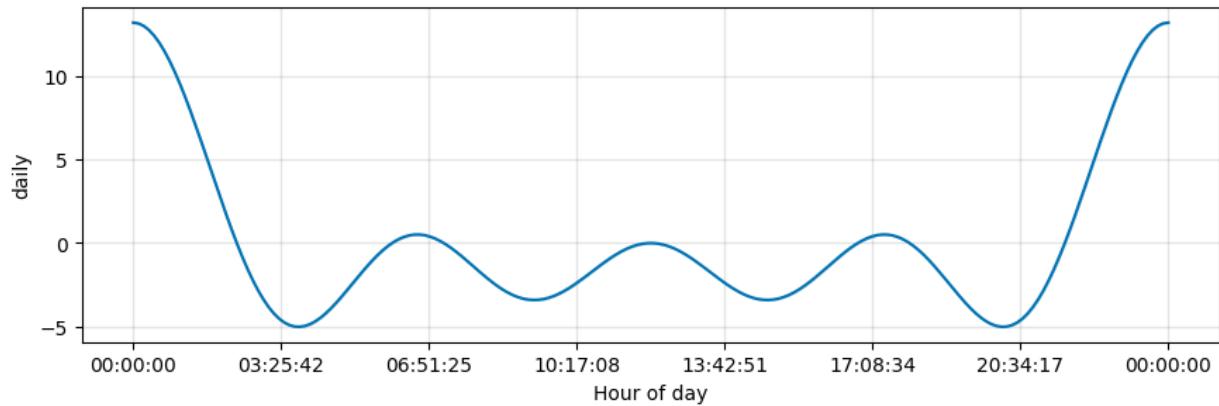
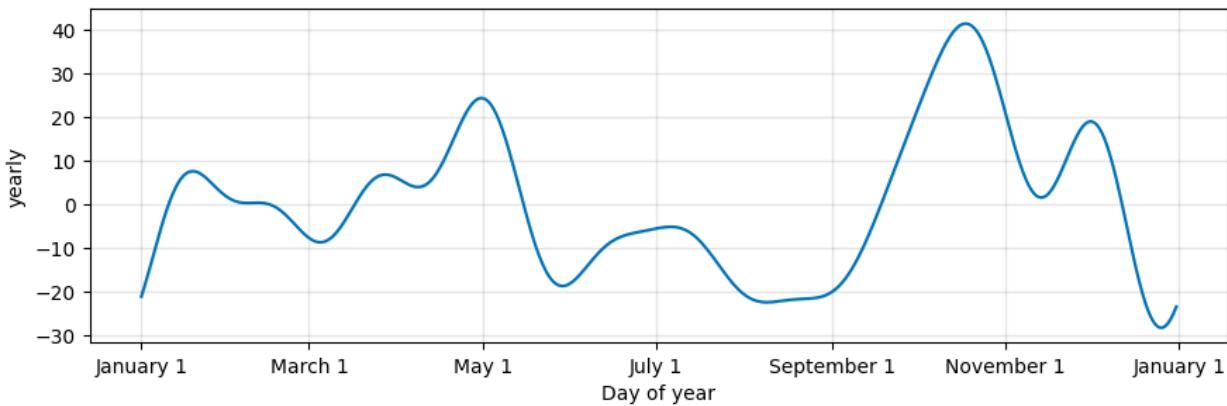
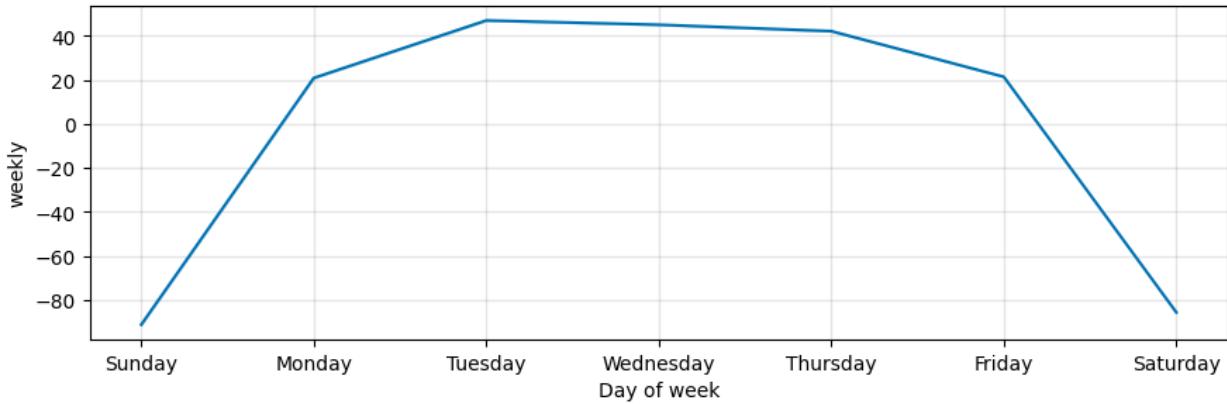
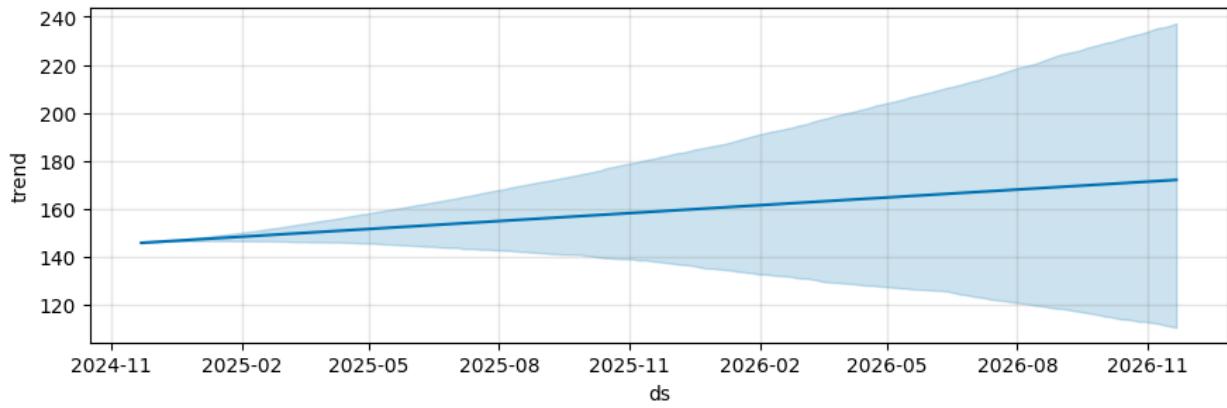
# Optional: Plot forecast components (trend, yearly, and weekly seasonality)
model_branches.plot_components(forecast_branches)
plt.show()

22:35:58 - cmdstanpy - INFO - Chain [1] start processing
22:35:58 - cmdstanpy - INFO - Chain [1] done processing

<Figure size 1000x600 with 0 Axes>

```





```

#Add your code for requirement 9.9 in this cell

import pandas as pd
import matplotlib.pyplot as plt
from prophet import Prophet

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter the data for 'pull_request' type
df_pulls = df_detailed[df_detailed['type'] == 'pull_request']

# Convert 'created_at' to datetime (assuming pulls have 'created_at' column)
df_pulls['created_at'] = pd.to_datetime(df_pulls['created_at'])

# Filter data between November 2022 and November 2024
start_date = '2022-11-01'
end_date = '2024-11-30'
df_pulls = df_pulls[(df_pulls['created_at'] >= start_date) &
(df_pulls['created_at'] <= end_date)]

# Extract the date part (ignoring time) and group by date to count pull requests
df_pulls['date'] = df_pulls['created_at'].dt.date
df_pulls_daily =
df_pulls.groupby('date').size().reset_index(name='pull_request_count')

# Prepare the data for Prophet (Prophet expects columns 'ds' for date and 'y' for the values to forecast)
prophet_df_pulls = df_pulls_daily.rename(columns={'date': 'ds',
'pull_request_count': 'y'})

# Initialize and fit the Prophet model
model_pulls = Prophet(daily_seasonality=True)
model_pulls.fit(prophet_df_pulls)

# Forecast the data based on existing dates
forecast_pulls = model_pulls.predict(prophet_df_pulls)

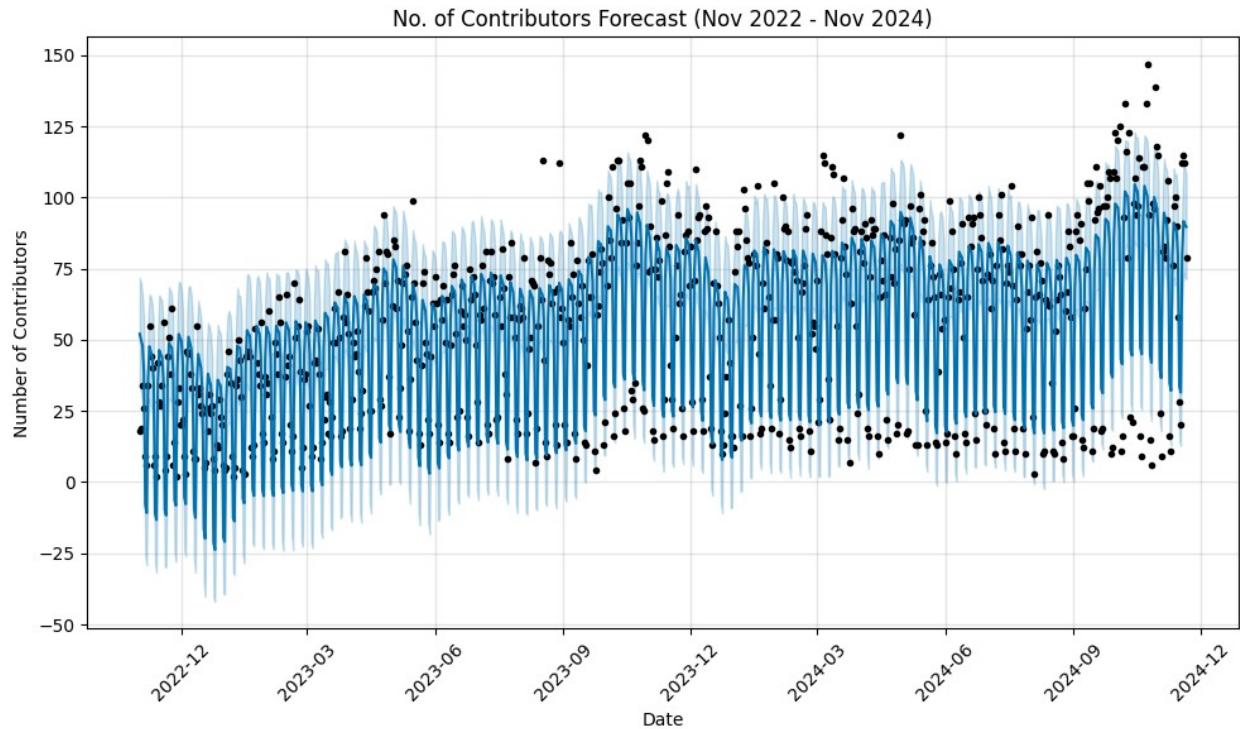
# Plot the pull requests forecast
plt.figure(figsize=(10, 6))
model_pulls.plot(forecast_pulls)
plt.title('No. of Contributors Forecast (Nov 2022 - Nov 2024)')
plt.xlabel('Date')
plt.ylabel('Number of Contributors')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("forecasting/Prophet/contributors_forecast.png")
plt.show()

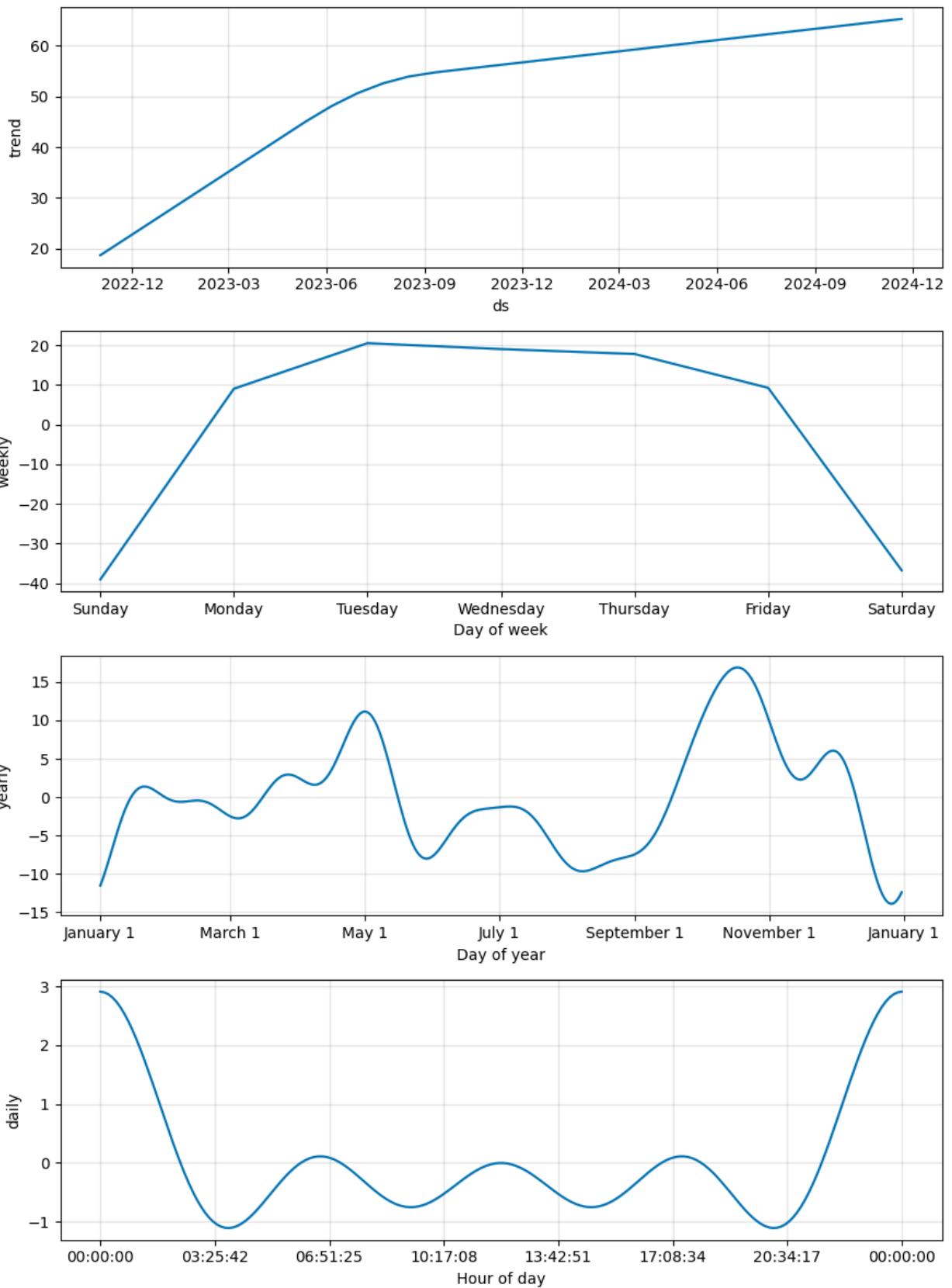
```

```
# Optional: Plot forecast components (trend, yearly, and weekly seasonality)
model_pulls.plot_components(forecast_pulls)
plt.show()
```

```
22:36:17 - cmdstanpy - INFO - Chain [1] start processing
22:36:18 - cmdstanpy - INFO - Chain [1] done processing
```

```
<Figure size 1000x600 with 0 Axes>
```





```

#Add your code for requirement 9.10 in this cell

import pandas as pd
import matplotlib.pyplot as plt
from prophet import Prophet

# Load the data
df_releases = pd.read_csv("github_all_data.csv")

# Assuming there is a 'release' or similar column to track releases
# (adjust the column name if necessary)
# Extract the date (assuming 'created_at' column contains timestamp)
df_releases['created_at'] = pd.to_datetime(df_releases['created_at'])

# Filter data between Nov 2022 and Nov 2024 (adjust as necessary)
start_date = '2022-11-01'
end_date = '2024-11-30'
df_releases = df_releases[(df_releases['created_at'] >= start_date) &
                           (df_releases['created_at'] <= end_date)]

# Filter for the 'release' column (assuming it exists and tracks
releases)
# Group by date and count the number of releases per day
df_releases['date'] = df_releases['created_at'].dt.date
df_releases_daily =
df_releases.groupby('date').size().reset_index(name='release_count')

# Prepare the data for Prophet (Prophet expects columns 'ds' for date
and 'y' for the values to forecast)
prophet_df_releases = df_releases_daily.rename(columns={'date': 'ds',
                                                       'release_count': 'y'})

# Initialize the Prophet model
model_releases = Prophet(daily_seasonality=True)

# Fit the model with the historical data
model_releases.fit(prophet_df_releases)

# Create a custom date range for the future predictions (next 2 years)
# For example, from the last date in the dataset, generate a list of
future dates
last_date = prophet_df_releases['ds'].max()
future_dates = pd.date_range(start=last_date, periods=365*2 + 1,
                             freq='D')[1:]

# Prepare the dataframe for future dates
future_df = pd.DataFrame(future_dates, columns=['ds'])

# Forecast the future values
forecast_releases = model_releases.predict(future_df)

```

```

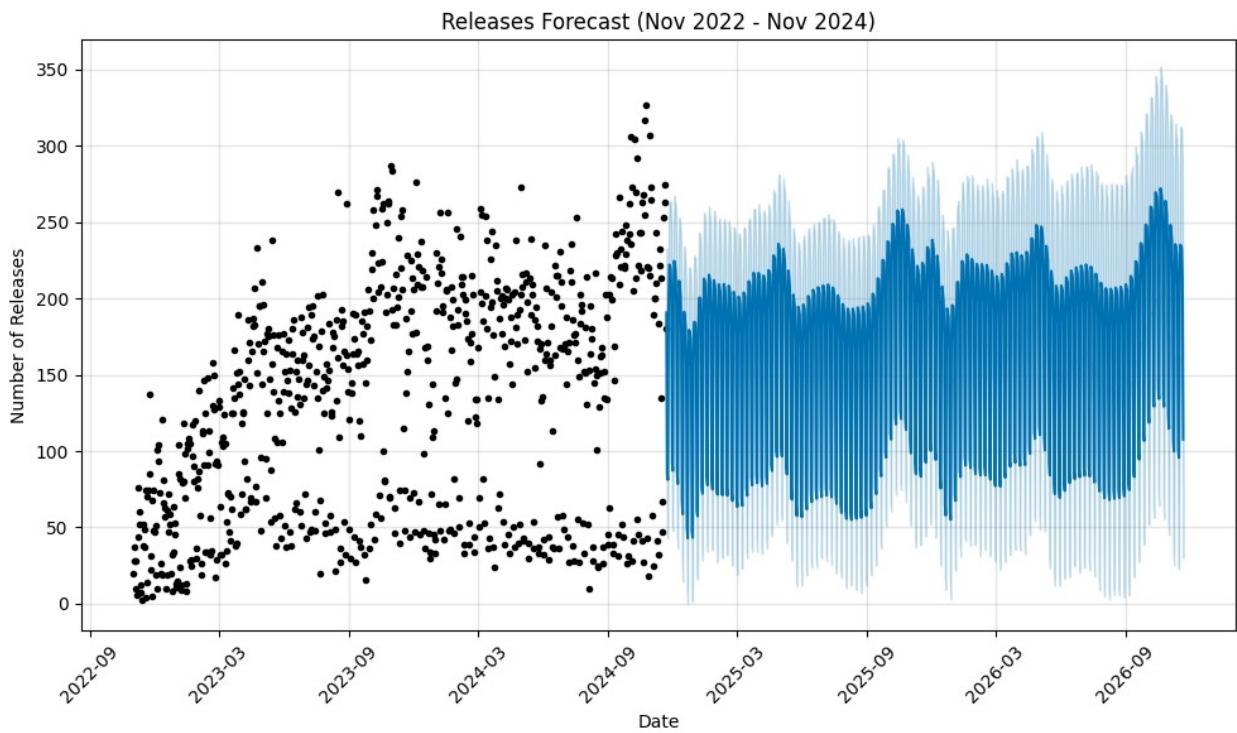
# Plot the releases forecast
plt.figure(figsize=(10, 6))
model_releases.plot(forecast_releases)
plt.title('Releases Forecast (Nov 2022 - Nov 2024)')
plt.xlabel('Date')
plt.ylabel('Number of Releases')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("forecasting/Prophet/releases_forecast.png")
plt.show()

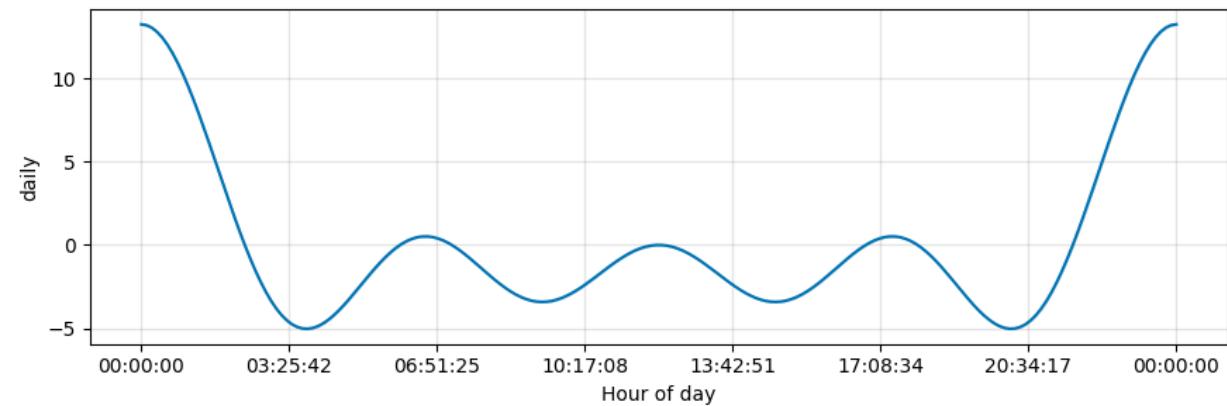
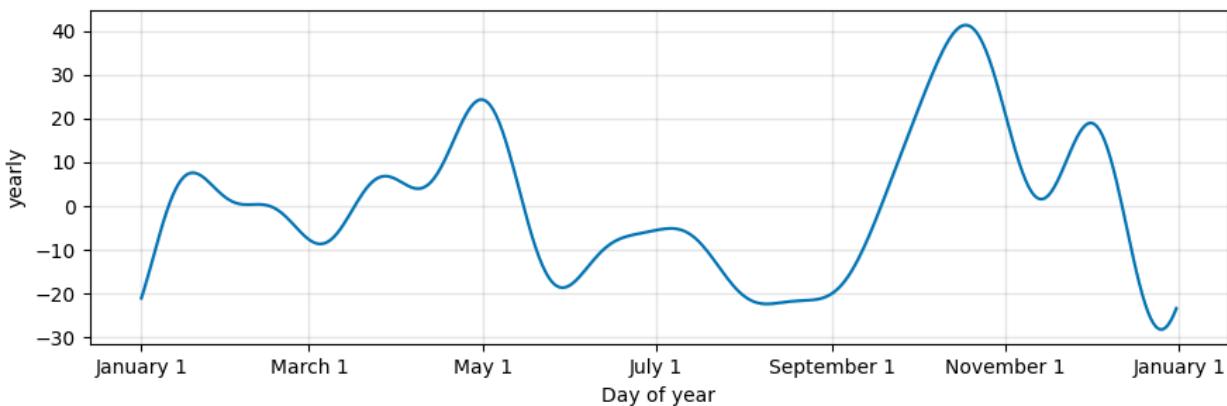
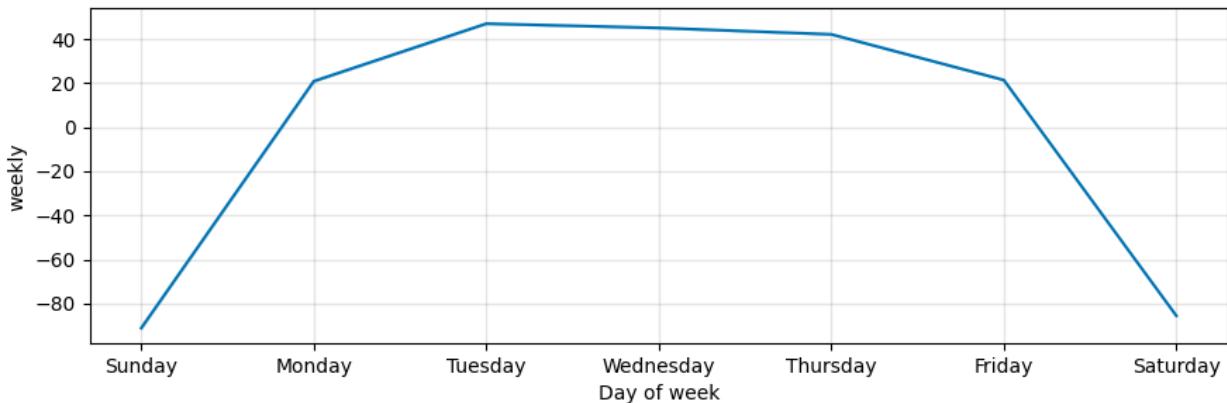
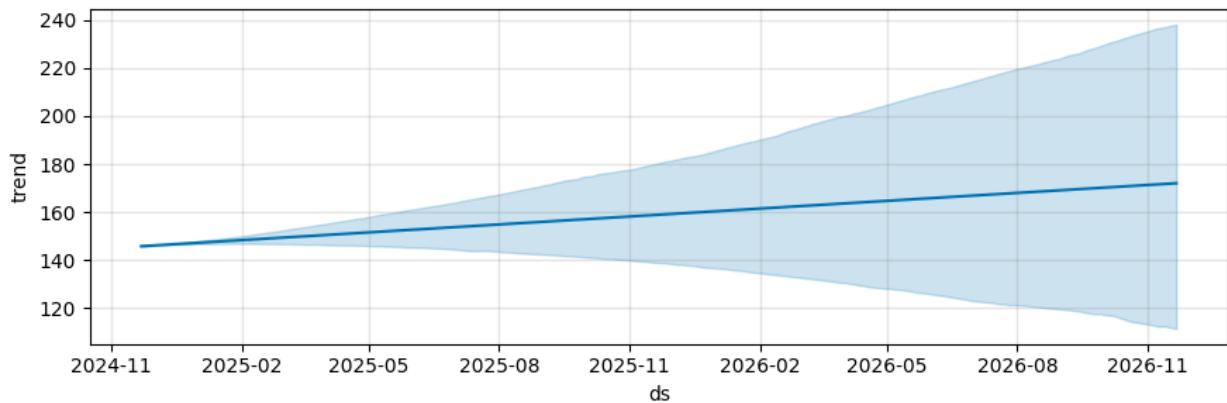
# Optional: Plot forecast components (trend, yearly, and weekly
# seasonality)
model_releases.plot_components(forecast_releases)
plt.show()

```

22:36:34 - cmdstanpy - INFO - Chain [1] start processing
22:36:34 - cmdstanpy - INFO - Chain [1] done processing

<Figure size 1000x600 with 0 Axes>





```

#Add your code for requirement 10.1 in this cell
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.tsa.arima.model import ARIMA

# Load the data
df_issues = pd.read_csv("github_all_data.csv")

# Assuming there is a 'created_at' column with timestamps for when
# issues were created
df_issues['created_at'] = pd.to_datetime(df_issues['created_at'])

# Filter data between Nov 2022 and Nov 2024 (adjust as necessary)
start_date = '2022-11-01'
end_date = '2024-11-30'
df_issues = df_issues[(df_issues['created_at'] >= start_date) &
(df_issues['created_at'] <= end_date)]

# Extract the day of the week (0 = Monday, 6 = Sunday)
df_issues['day_of_week'] = df_issues['created_at'].dt.dayofweek

# Group by 'day_of_week' and count the number of issues created on
# each day
issues_by_day =
df_issues.groupby('day_of_week').size().reset_index(name='issue_count')

# Plot the issue count by day of the week
plt.figure(figsize=(10, 6))
plt.bar(issues_by_day['day_of_week'], issues_by_day['issue_count'],
color='skyblue')
plt.xlabel('Day of the Week')
plt.ylabel('Number of Issues')
plt.title('Issues Created by Day of the Week (Nov 2022 - Nov 2024)')
plt.xticks(issues_by_day['day_of_week'], ['Mon', 'Tue', 'Wed', 'Thu',
'Fri', 'Sat', 'Sun'])
plt.show()

# Prepare the time series data for ARIMA
# Assume we want to predict the maximum number of issues on the
# weekdays (0 = Monday, ..., 6 = Sunday)
# Reshape the data to fit the ARIMA model
# The day of the week is categorical but we're treating the counts as
# time series for forecasting

# Aggregate the issues count on a daily basis
df_issues_daily = df_issues.groupby(df_issues['created_at'].dt.date)[
'day_of_week'].value_counts().unstack(fill_value=0)

# Forecasting for the next period using ARIMA model

```

```

# We need to forecast for future days of the week based on historical data

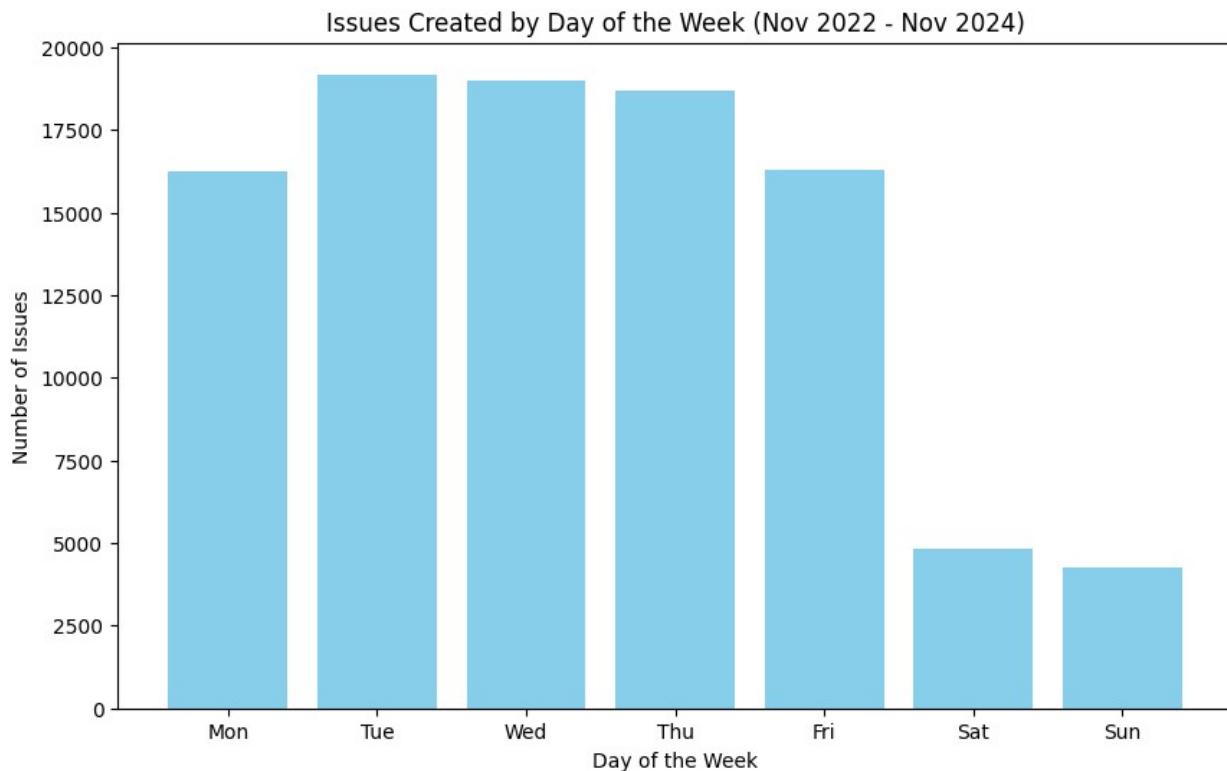
# Here we demonstrate a simple ARIMA model (p, d, q) for forecasting
model = ARIMA(df_issues_daily.sum(axis=0), order=(5, 1, 0)) # Change the order (p, d, q) as needed
model_fit = model.fit()

# Make predictions for the next 7 days (for each day of the week)
forecast = model_fit.forecast(steps=7)

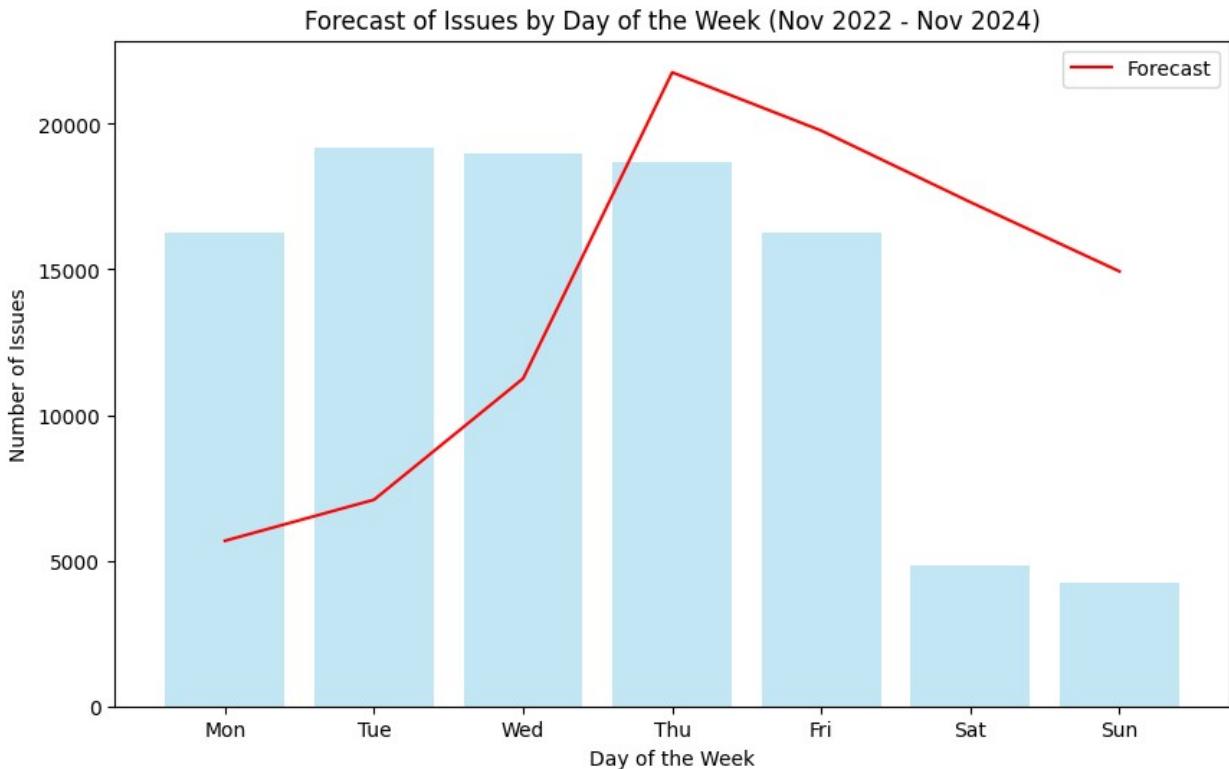
# Plot the forecast
plt.figure(figsize=(10, 6))
plt.plot(range(7), forecast, label="Forecast", color='red')
plt.bar(issues_by_day['day_of_week'], issues_by_day['issue_count'],
color='skyblue', alpha=0.5)
plt.xticks(range(7), ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat',
'Sun'])
plt.xlabel('Day of the Week')
plt.ylabel('Number of Issues')
plt.title('Forecast of Issues by Day of the Week (Nov 2022 - Nov 2024)')
plt.legend()
plt.savefig("forecasting/StatsModel/max_number_of_issues_created.png")
plt.show()

# Identify the day with the maximum number of issues
max_day_of_week = forecast.argmax() # Get the day of the week with max forecast
print(f"The day of the week with the maximum number of issues forecasted is: {[ 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'][max_day_of_week]}")

```



```
c:\Users\soham\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\base\model.py:607: ConvergenceWarning: Maximum
Likelihood optimization failed to converge. Check mle_retrvals
warnings.warn("Maximum Likelihood optimization failed to "
```



The day of the week with the maximum number of issues forecasted is:
Thu

#Add your code for requirement 10.2 in this cell

```
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.tsa.arima.model import ARIMA

# Load the data
df_issues = pd.read_csv("github_all_data.csv")

# Assuming there is a 'closed_at' column with timestamps for when
# issues were closed
df_issues['closed_at'] = pd.to_datetime(df_issues['closed_at'],
                                         errors='coerce')

# Filter data between Nov 2022 and Nov 2024 (adjust as necessary)
start_date = '2022-11-01'
end_date = '2024-11-30'
df_issues = df_issues[(df_issues['closed_at'] >= start_date) &
                      (df_issues['closed_at'] <= end_date)]

# Extract the day of the week (0 = Monday, 6 = Sunday) from the
# 'closed_at' column
```

```

df_issues['day_of_week_closed'] = df_issues['closed_at'].dt.dayofweek

# Group by 'day_of_week_closed' and count the number of issues closed
# on each day
issues_closed_by_day =
df_issues.groupby('day_of_week_closed').size().reset_index(name='close
d_count')

# Plot the issue closure count by day of the week
plt.figure(figsize=(10, 6))
plt.bar(issues_closed_by_day['day_of_week_closed'],
issues_closed_by_day['closed_count'], color='lightgreen')
plt.xlabel('Day of the Week')
plt.ylabel('Number of Issues Closed')
plt.title('Issues Closed by Day of the Week (Nov 2022 - Nov 2024)')
plt.xticks(issues_closed_by_day['day_of_week_closed'], ['Mon', 'Tue',
'Wed', 'Thu', 'Fri', 'Sat', 'Sun'])
plt.show()

# Prepare the time series data for ARIMA model (Sum of issues closed
per day of the week)
# Aggregate the closed count data on a daily basis (by day of the
week)
df_issues_daily_closed =
df_issues.groupby(df_issues['closed_at'].dt.date)
['day_of_week_closed'].value_counts().unstack(fill_value=0)

# Forecasting for the next period using ARIMA model
# We will forecast the number of issues closed for the next 7 days of
the week
model = ARIMA(issues_closed_by_day['closed_count'], order=(5, 1, 0))
# ARIMA(p,d,q) model (adjust p, d, q as needed)
model_fit = model.fit()

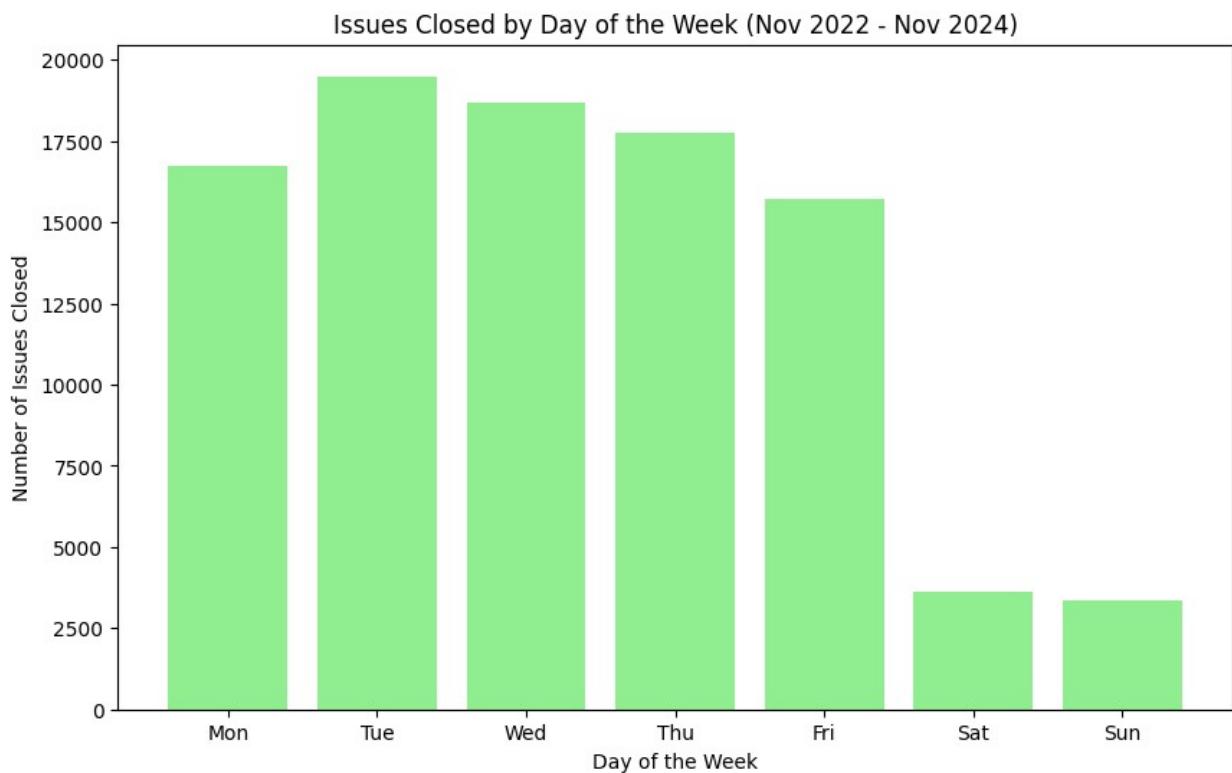
# Make predictions for the next 7 days (for each day of the week)
forecast = model_fit.forecast(steps=7)

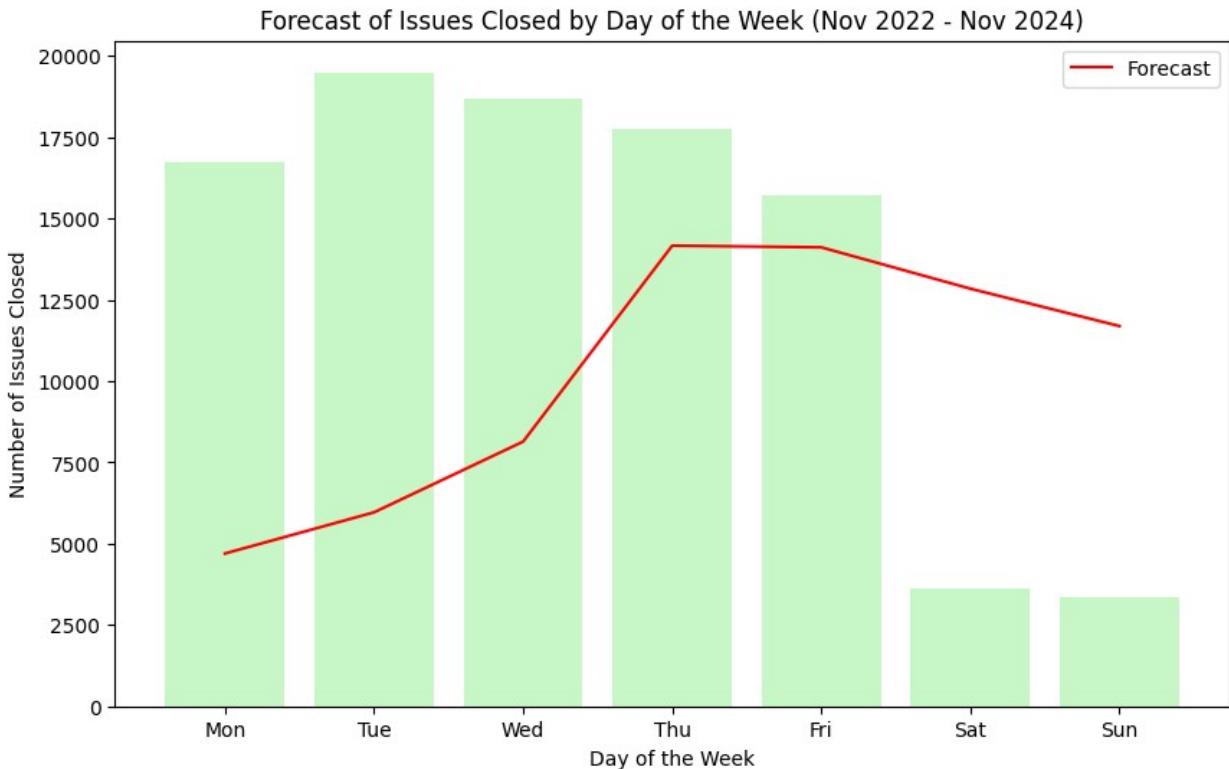
# Plot the forecast
plt.figure(figsize=(10, 6))
plt.plot(range(7), forecast, label="Forecast", color='red')
plt.bar(issues_closed_by_day['day_of_week_closed'],
issues_closed_by_day['closed_count'], color='lightgreen', alpha=0.5)
plt.xticks(range(7), ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat',
'Sun'])
plt.xlabel('Day of the Week')
plt.ylabel('Number of Issues Closed')
plt.title('Forecast of Issues Closed by Day of the Week (Nov 2022 -
Nov 2024)')
plt.legend()
plt.savefig("forecasting/StatsModel/max_number_of_issues_closed.png")

```

```
plt.show()

# Identify the day with the maximum number of issues closed
max_day_of_week = forecast.argmax() # Get the day of the week with
# max forecast
print(f"The day of the week with the maximum number of issues closed
forecasted is: {[ 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
[max_day_of_week]}")
```





The day of the week with the maximum number of issues closed forecasted is: Thu

```
#Add your code for requirement 10.3 in this cell
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.tsa.arima.model import ARIMA

# Load the data
df_issues = pd.read_csv("github_all_data.csv")

# Assuming there is a 'closed_at' column with timestamps for when
# issues were closed
df_issues['closed_at'] = pd.to_datetime(df_issues['closed_at'],
                                         errors='coerce')

# Filter data between Nov 2022 and Nov 2024 (adjust as necessary)
start_date = '2022-11-01'
end_date = '2024-11-30'
df_issues = df_issues[(df_issues['closed_at'] >= start_date) &
                      (df_issues['closed_at'] <= end_date)]

# Extract the month from the 'closed_at' column
df_issues['month_closed'] = df_issues['closed_at'].dt.month
```

```

# Group by 'month_closed' and count the number of issues closed in each month
issues_closed_by_month =
df_issues.groupby('month_closed').size().reset_index(name='closed_count')

# Plot the issue closure count by month
plt.figure(figsize=(10, 6))
plt.bar(issues_closed_by_month['month_closed'],
issues_closed_by_month['closed_count'], color='lightcoral')
plt.xlabel('Month')
plt.ylabel('Number of Issues Closed')
plt.title('Issues Closed by Month (Nov 2022 - Nov 2024)')
plt.xticks(issues_closed_by_month['month_closed'], ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])
plt.show()

# Prepare the time series data for ARIMA model (Sum of issues closed per month)
# Aggregate the closed count data on a monthly basis
df_issues_monthly_closed =
df_issues.groupby(df_issues['closed_at'].dt.to_period('M'))['month_closed'].value_counts().unstack(fill_value=0)

# Forecasting for the next period using ARIMA model
# We will forecast the number of issues closed for the next 12 months
model = ARIMA(issues_closed_by_month['closed_count'], order=(5, 1, 0))
# ARIMA(p,d,q) model (adjust p, d, q as needed)
model_fit = model.fit()

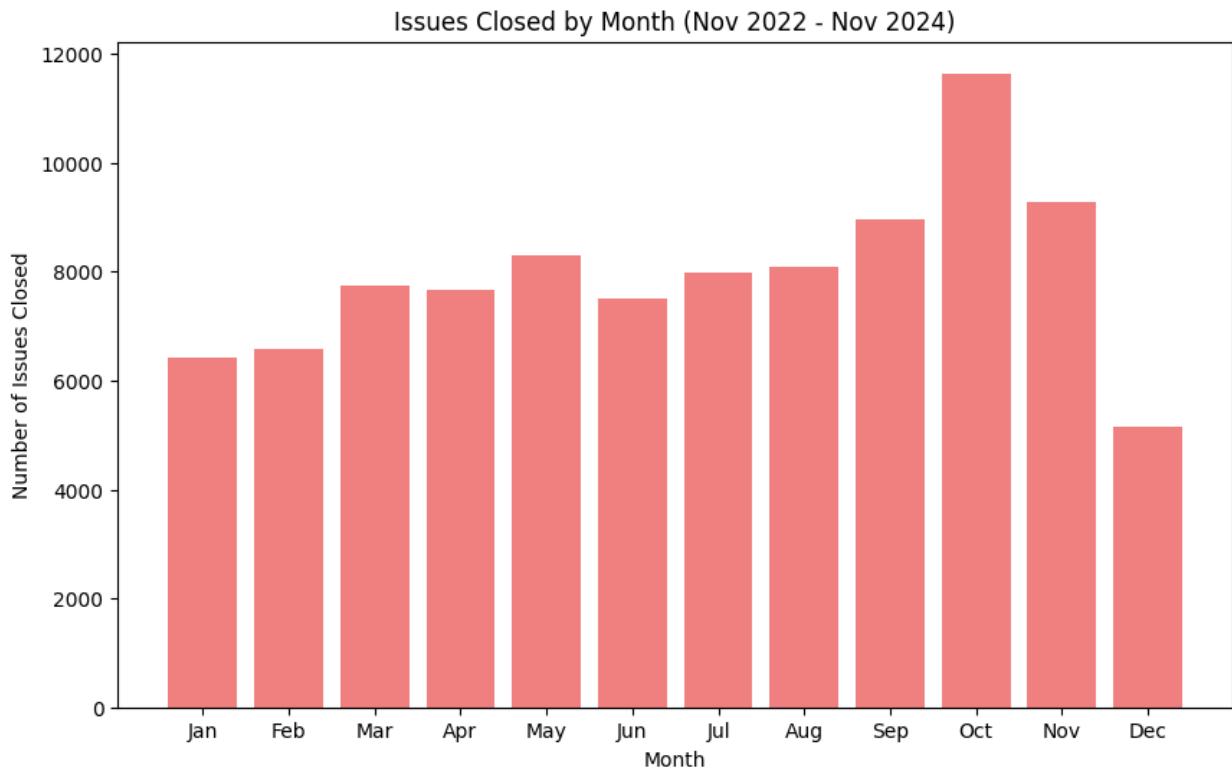
# Make predictions for the next 12 months
forecast = model_fit.forecast(steps=12)

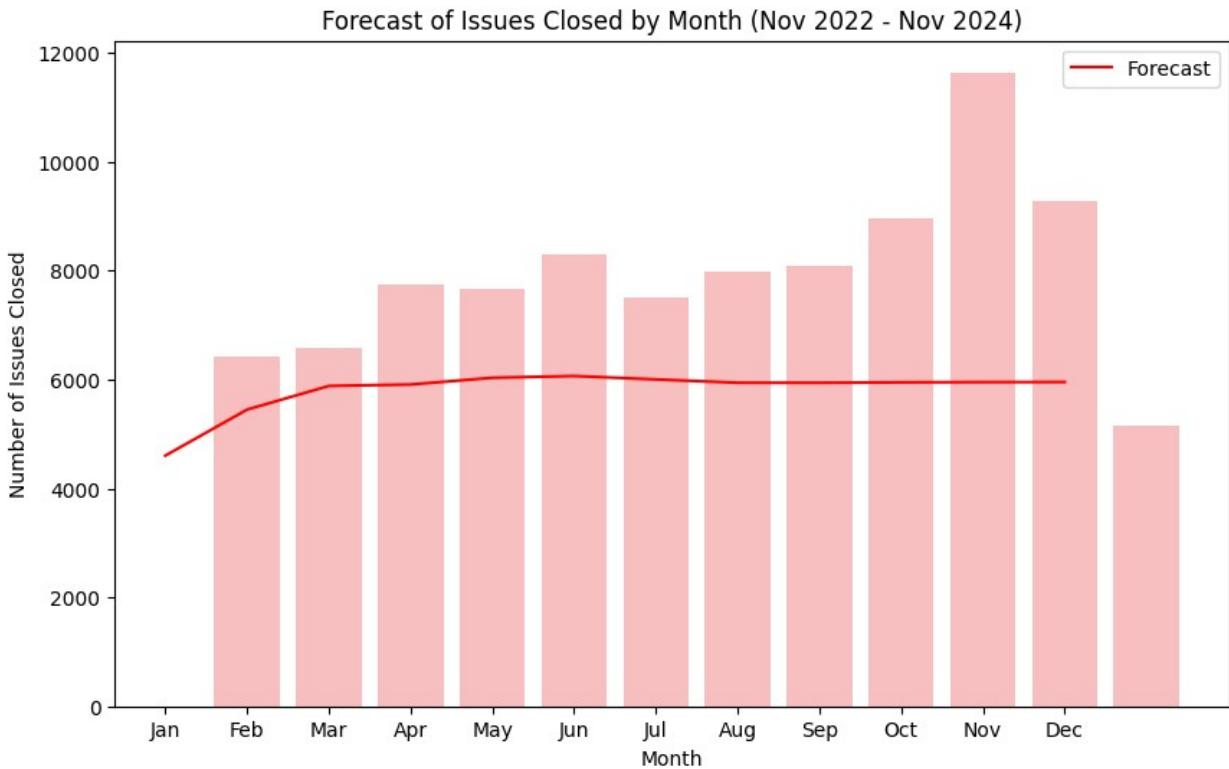
# Plot the forecast
plt.figure(figsize=(10, 6))
plt.plot(range(12), forecast, label="Forecast", color='red')
plt.bar(issues_closed_by_month['month_closed'],
issues_closed_by_month['closed_count'], color='lightcoral', alpha=0.5)
plt.xticks(range(12), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])
plt.xlabel('Month')
plt.ylabel('Number of Issues Closed')
plt.title('Forecast of Issues Closed by Month (Nov 2022 - Nov 2024)')
plt.legend()
plt.savefig("forecasting/StatsModel/max_number_of_issues_closed_in_month.png")
plt.show()

# Identify the month with the maximum number of issues closed
max_month = forecast.argmax() # Get the month with max forecast

```

```
print(f"The month with the maximum number of issues closed forecasted  
is: {[('Jan', 6450), ('Feb', 6650), ('Mar', 7750), ('Apr', 7600), ('May', 8350),  
('Jun', 7500), ('Jul', 8000), ('Aug', 8100), ('Sep', 9000), ('Oct', 11600), ('Nov', 9250),  
('Dec', 5150)][max_month]}")
```





The month with the maximum number of issues closed forecasted is: Jun

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# Load the data
df_issues = pd.read_csv("github_all_data.csv")

# Assuming there is a 'created_at' column with timestamps for when
# issues were created
df_issues['created_at'] = pd.to_datetime(df_issues['created_at'],
                                         errors='coerce')

# Filter data between Nov 2022 and Nov 2024 (adjust as necessary)
start_date = '2022-11-01'
end_date = '2024-11-30'
df_issues = df_issues[(df_issues['created_at'] >= start_date) &
                      (df_issues['created_at'] <= end_date)]

# Group by month and count the number of issues created
df_issues['month_created'] = df_issues['created_at'].dt.to_period('M')
# Use monthly periods
issues_created_by_month =
df_issues.groupby('month_created').size().reset_index(name='created_count')
```

```

# Ensure the data is sorted by date
issues_created_by_month =
issues_created_by_month.sort_values(by='month_created')

# Prepare the time series data
created_time_series = issues_created_by_month['created_count'].values

# Fit the ARIMA model
model = ARIMA(created_time_series, order=(5, 1, 0)) # Adjust p, d, q
as needed
model_fit = model.fit()

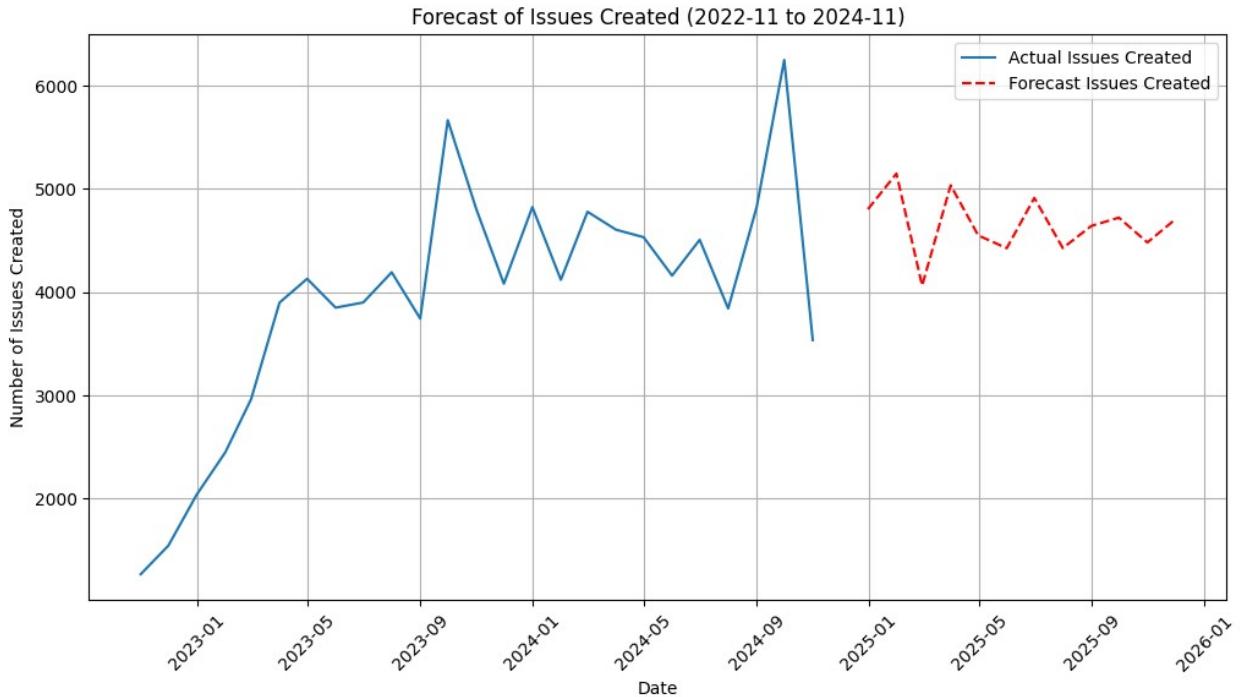
# Forecast the number of issues created for the next 12 months
forecast_steps = 12
forecast = model_fit.forecast(steps=forecast_steps)

# Generate a range of future dates for plotting
future_dates =
pd.date_range(start=issues_created_by_month['month_created'].iloc[-1].start_time,
              periods=forecast_steps + 1, freq='M')[1:]

# Plot the actual data and forecast
plt.figure(figsize=(12, 6))
plt.plot(issues_created_by_month['month_created'].dt.to_timestamp(),
         created_time_series, label="Actual Issues Created")
plt.plot(future_dates, forecast, label="Forecast Issues Created",
         linestyle='--', color='red')
plt.xlabel("Date")
plt.ylabel("Number of Issues Created")
plt.title("Forecast of Issues Created (2022-11 to 2024-11)")
plt.xticks(rotation=45)
plt.legend()
plt.grid()
plt.savefig("forecasting/StatsModel/created_issues_forecast.png")
plt.show()

# Print forecasted values
forecast_df = pd.DataFrame({
    "Date": future_dates,
    "Forecasted Issues Created": forecast
})
print(forecast_df)

```



	Date	Forecasted Issues Created
0	2024-12-31	4803.673349
1	2025-01-31	5149.128750
2	2025-02-28	4063.048186
3	2025-03-31	5035.458104
4	2025-04-30	4549.754794
5	2025-05-31	4425.611624
6	2025-06-30	4913.479237
7	2025-07-31	4426.535187
8	2025-08-31	4641.597636
9	2025-09-30	4721.909912
10	2025-10-31	4481.387153
11	2025-11-30	4702.825559

#Add your code for requirement 10.5 in this cell

```

import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# Load the data
df_issues = pd.read_csv("github_all_data.csv")

# Assuming there is a 'closed_at' column with timestamps for when
# issues were closed
df_issues['closed_at'] = pd.to_datetime(df_issues['closed_at'],
errors='coerce')

```

```

# Filter data between Nov 2022 and Nov 2024
start_date = '2022-11-01'
end_date = '2024-11-30'
df_issues = df_issues[(df_issues['closed_at'] >= start_date) &
(df_issues['closed_at'] <= end_date)]

# Group by month and count the number of issues closed
df_issues['month_closed'] = df_issues['closed_at'].dt.to_period('M')
# Monthly periods
issues_closed_by_month =
df_issues.groupby('month_closed').size().reset_index(name='closed_count')

# Ensure the data is sorted by date
issues_closed_by_month =
issues_closed_by_month.sort_values(by='month_closed')

# Prepare the time series data
closed_time_series = issues_closed_by_month['closed_count'].values

# Fit the ARIMA model
model = ARIMA(closed_time_series, order=(5, 1, 0)) # Adjust p, d, q as needed
model_fit = model.fit()

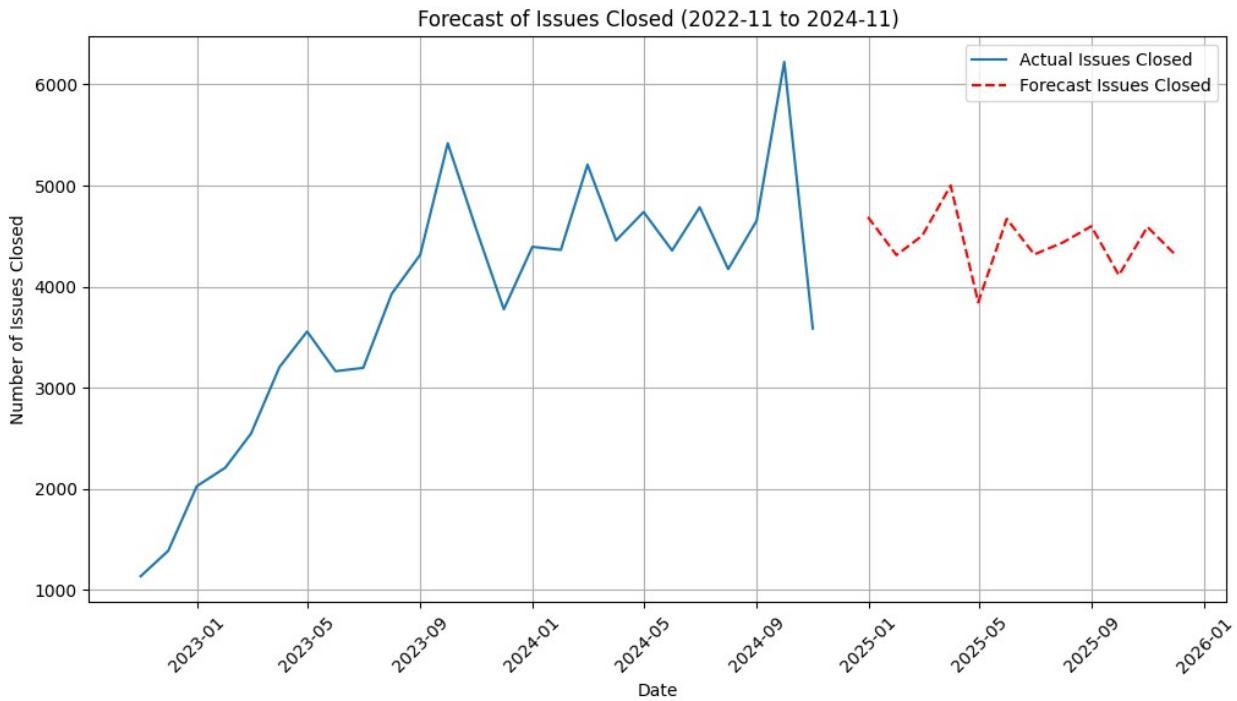
# Forecast the number of closed issues for the next 12 months
forecast_steps = 12
forecast = model_fit.forecast(steps=forecast_steps)

# Generate a range of future dates for plotting
future_dates =
pd.date_range(start=issues_closed_by_month['month_closed'].iloc[-1].start_time,
              periods=forecast_steps + 1, freq='M')[1:]

# Plot the actual data and forecast
plt.figure(figsize=(12, 6))
plt.plot(issues_closed_by_month['month_closed'].dt.to_timestamp(),
closed_time_series, label="Actual Issues Closed")
plt.plot(future_dates, forecast, label="Forecast Issues Closed",
linestyle='--', color='red')
plt.xlabel("Date")
plt.ylabel("Number of Issues Closed")
plt.title("Forecast of Issues Closed (2022-11 to 2024-11)")
plt.xticks(rotation=45)
plt.legend()
plt.grid()
plt.savefig("forecasting/StatsModel/closed_issues_forecast.png")
plt.show()

```

```
# Print forecasted values
forecast_df = pd.DataFrame({
    "Date": future_dates,
    "Forecasted Issues Closed": forecast
})
print(forecast_df)
```



	Date	Forecasted Issues Closed
0	2024-12-31	4688.696516
1	2025-01-31	4311.873660
2	2025-02-28	4502.395555
3	2025-03-31	5000.187484
4	2025-04-30	3843.591208
5	2025-05-31	4670.214251
6	2025-06-30	4318.143635
7	2025-07-31	4435.691239
8	2025-08-31	4597.198920
9	2025-09-30	4112.097784
10	2025-10-31	4590.599248
11	2025-11-30	4318.158305

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# Load the detailed data
```

```

df_detailed = pd.read_csv("github_all_data.csv")

# Filter the data for 'pull_request' type and 'closed' state
df_pulls = df_detailed[df_detailed['type'] == 'pull_request']
df_pulls = df_pulls[df_pulls['state'] == 'closed']

# Convert 'created_at' to datetime
df_pulls['created_at'] = pd.to_datetime(df_pulls['created_at'])

# Filter data between November 2022 and November 2024
start_date = '2022-11-01'
end_date = '2024-11-30'
df_pulls = df_pulls[(df_pulls['created_at'] >= start_date) &
(df_pulls['created_at'] <= end_date)]

# Group by date and count pull requests
df_pulls['date'] = df_pulls['created_at'].dt.date
df_pulls_daily =
df_pulls.groupby('date').size().reset_index(name='pull_request_count')

# Prepare the time series data
pulls_time_series = df_pulls_daily['pull_request_count'].values

# Fit the ARIMA model
model = ARIMA(pulls_time_series, order=(5, 1, 0)) # Adjust p, d, q as needed
model_fit = model.fit()

# Forecast the daily pull requests for the next 30 days (or as needed)
forecast_steps = 365 # Forecast for one year
forecast = model_fit.forecast(steps=forecast_steps)

# Generate future dates for plotting
last_date = df_pulls_daily['date'].iloc[-1]
future_dates = pd.date_range(start=last_date, periods=forecast_steps + 1, freq='D')[1:]

# Plot the actual and forecasted pull requests
plt.figure(figsize=(12, 6))
plt.plot(df_pulls_daily['date'], pulls_time_series, label="Actual Pull Requests")
plt.plot(future_dates, forecast, label="Forecasted Pull Requests",
linestyle='--', color='red')
plt.xlabel("Date")
plt.ylabel("Number of Pull Requests")
plt.title("Pull Requests Forecast (Nov 2022 to Nov 2024)")
plt.xticks(rotation=45)
plt.legend()
plt.grid()
plt.savefig("forecasting/StatsModel/pulls_forecast.png")

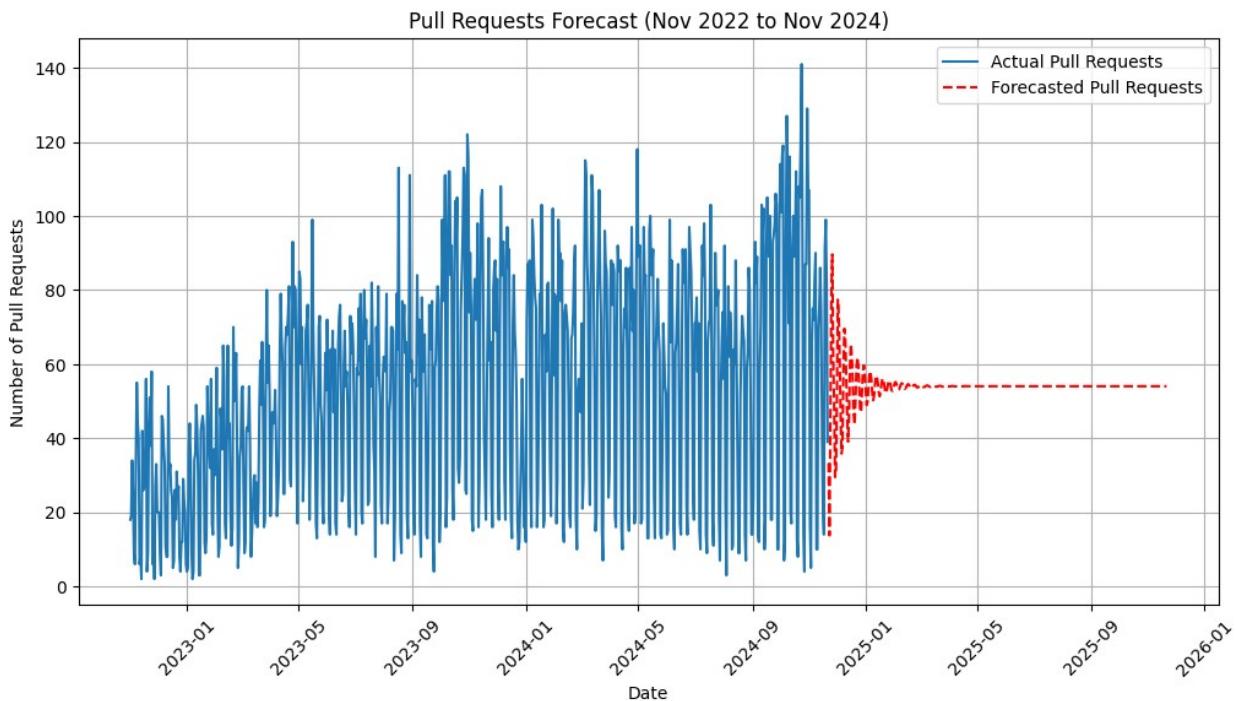
```

```

plt.show()

# Print forecasted values
forecast_df = pd.DataFrame({
    "Date": future_dates,
    "Forecasted Pull Requests": forecast
})
print(forecast_df)

```



	Date	Forecasted Pull Requests
0	2024-11-22	33.302323
1	2024-11-23	13.771657
2	2024-11-24	45.582651
3	2024-11-25	78.298471
4	2024-11-26	89.637294
..
360	2025-11-17	54.036626
361	2025-11-18	54.036625
362	2025-11-19	54.036625
363	2025-11-20	54.036626
364	2025-11-21	54.036626

[365 rows x 2 columns]

```

import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

```

```

from datetime import datetime

# Load the data
df_commits = pd.read_csv("github_all_data.csv")

# Extract commit date (assuming the 'created_at' column contains
# commit timestamp)
df_commits['created_at'] = pd.to_datetime(df_commits['created_at'])

# Filter commits from November 2022 to November 2024
start_date = '2022-11-01'
end_date = '2024-11-30'
df_commits = df_commits[(df_commits['created_at'] >= start_date) &
(df_commits['created_at'] <= end_date)]

# Group by date and count the number of commits per day
df_commits['date'] = df_commits['created_at'].dt.date
df_commits_daily =
df_commits.groupby('date').size().reset_index(name='commit_count')

# Convert 'date' to datetime and set as index
df_commits_daily['date'] = pd.to_datetime(df_commits_daily['date'])
df_commits_daily.set_index('date', inplace=True)

# Plot the original data
plt.figure(figsize=(10, 6))
plt.plot(df_commits_daily.index, df_commits_daily['commit_count'],
label='Commit Count')
plt.xlabel('Date')
plt.ylabel('Number of Commits')
plt.title('Commits Over Time (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.show()

# Fit the ARIMA model
# The (p, d, q) parameters can be tuned based on the dataset's
# characteristics.
model = ARIMA(df_commits_daily['commit_count'], order=(5, 1, 2))
fitted_model = model.fit()

# Forecast commits from November 2024 to November 2025
forecast_steps = 365 # Forecast for one year (adjust as needed)
forecast = fitted_model.get_forecast(steps=forecast_steps)

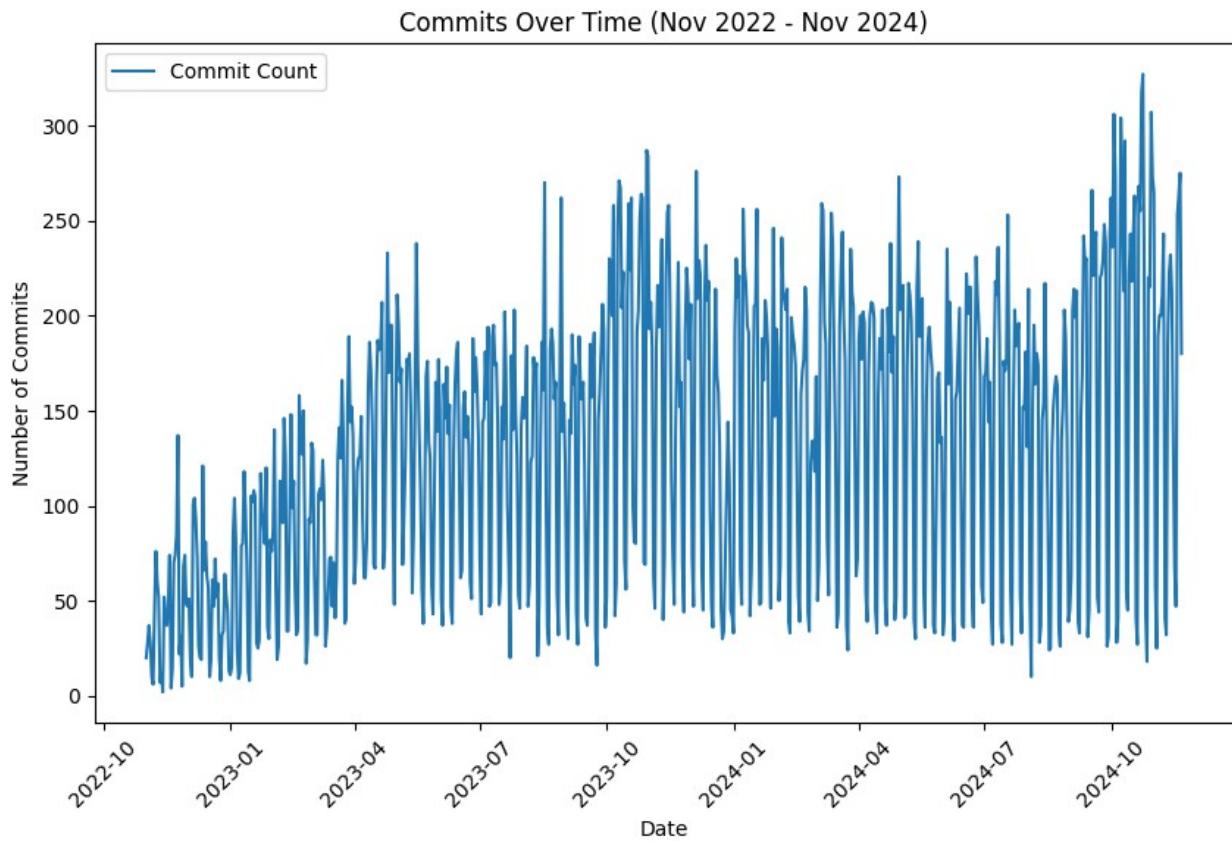
# Extract forecasted values and confidence intervals
forecast_index = pd.date_range(start=df_commits_daily.index[-1] +
pd.Timedelta(days=1), periods=forecast_steps, freq='D')
forecast_values = forecast.predicted_mean
forecast_ci = forecast.conf_int()

```

```

# Plot the forecast
plt.figure(figsize=(12, 6))
plt.plot(df_commits_daily.index, df_commits_daily['commit_count'],
label='Actual Commits', color='blue')
plt.plot(forecast_index, forecast_values, label='Forecasted Commits',
color='orange', linestyle='--')
plt.fill_between(forecast_index, forecast_ci.iloc[:, 0],
forecast_ci.iloc[:, 1], color='orange', alpha=0.2, label='Confidence
Interval')
plt.xlabel('Date')
plt.ylabel('Number of Commits')
plt.title('Commits Forecast (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.savefig("forecasting/StatsModel/commits_forecast.png")
plt.show()

```



```

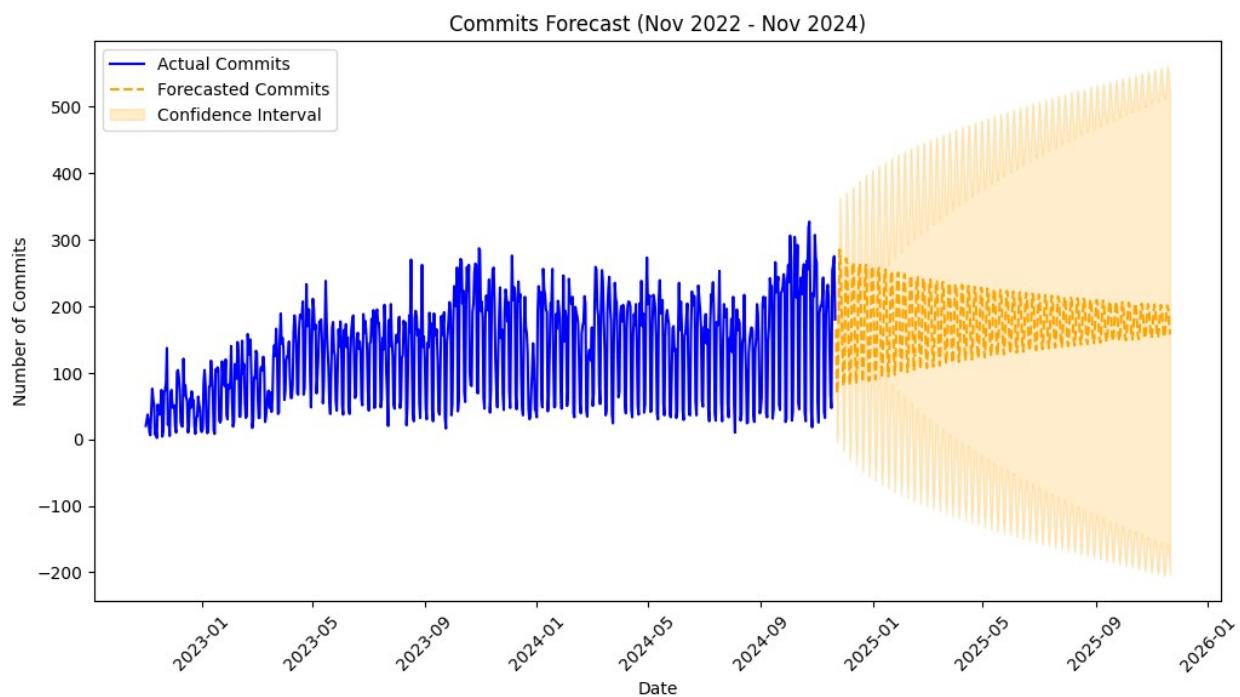
c:\Users\soham\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: No
frequency information was provided, so inferred frequency D will be
used.
    self._init_dates(dates, freq)

```

```

c:\Users\soham\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: No
frequency information was provided, so inferred frequency D will be
used.
    self._init_dates(dates, freq)
c:\Users\soham\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: No
frequency information was provided, so inferred frequency D will be
used.
    self._init_dates(dates, freq)
c:\Users\soham\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\base\model.py:607: ConvergenceWarning: Maximum
Likelihood optimization failed to converge. Check mle_retvals
warnings.warn("Maximum Likelihood optimization failed to "

```



```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# Load the detailed data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter the data for 'pull_request' type and 'closed' state
df_pulls = df_detailed[df_detailed['type'] == 'pull_request']
df_pulls = df_pulls[df_pulls['state'] == 'closed']

# Convert 'created_at' to datetime

```

```

df_pulls['created_at'] = pd.to_datetime(df_pulls['created_at'])

# Filter data between November 2022 and November 2024
start_date = '2022-11-01'
end_date = '2024-11-30'
df_pulls = df_pulls[(df_pulls['created_at'] >= start_date) &
(df_pulls['created_at'] <= end_date)]

# Group by date and count pull requests
df_pulls['date'] = df_pulls['created_at'].dt.date
df_pulls_daily =
df_pulls.groupby('date').size().reset_index(name='pull_request_count')

# Prepare the time series data
pulls_time_series = df_pulls_daily['pull_request_count'].values

# Fit the ARIMA model
model = ARIMA(pulls_time_series, order=(5, 1, 0)) # Adjust p, d, q as needed
model_fit = model.fit()

# Forecast the daily pull requests for the next 30 days (or as needed)
forecast_steps = 365 # Forecast for one year
forecast = model_fit.forecast(steps=forecast_steps)

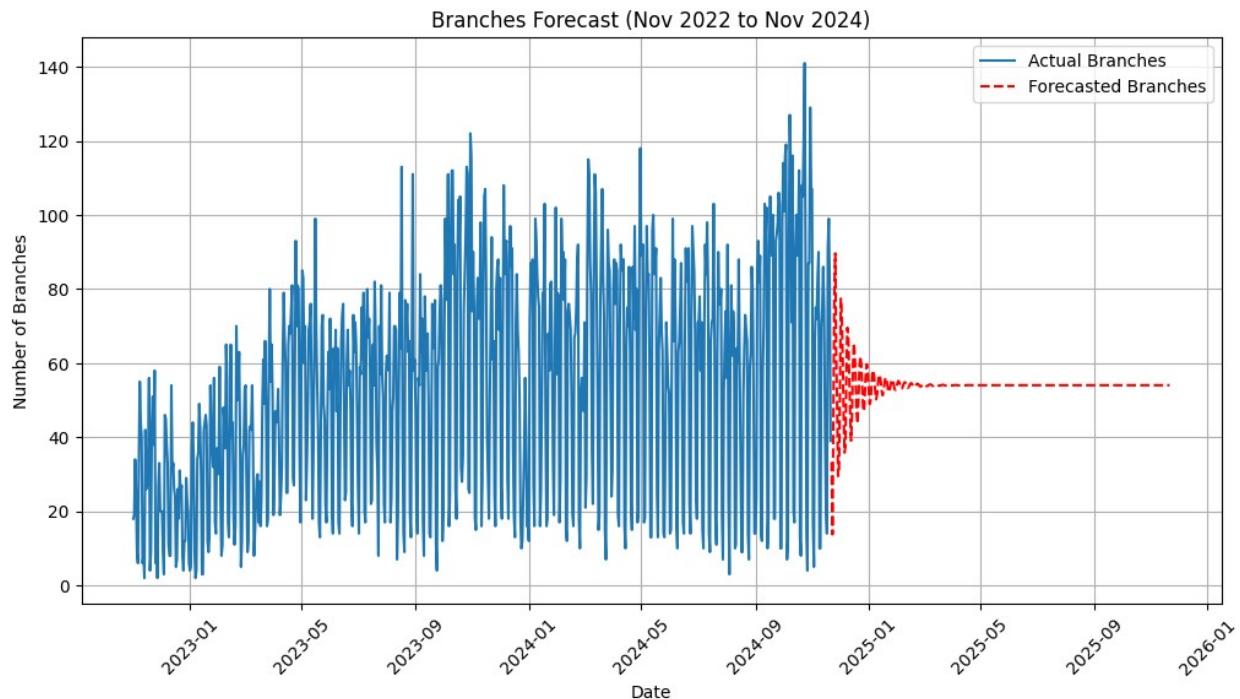
# Generate future dates for plotting
last_date = df_pulls_daily['date'].iloc[-1]
future_dates = pd.date_range(start=last_date, periods=forecast_steps + 1, freq='D')[1:]

# Plot the actual and forecasted pull requests
plt.figure(figsize=(12, 6))
plt.plot(df_pulls_daily['date'], pulls_time_series, label="Actual Branches")
plt.plot(future_dates, forecast, label="Forecasted Branches",
linestyle='--', color='red')
plt.xlabel("Date")
plt.ylabel("Number of Branches")
plt.title("Branches Forecast (Nov 2022 to Nov 2024)")
plt.xticks(rotation=45)
plt.legend()
plt.grid()
plt.savefig("forecasting/StatsModel/branches_forecast.png")
plt.show()

# Print forecasted values
forecast_df = pd.DataFrame({
    "Date": future_dates,
    "Forecasted Branches": forecast
})

```

```
})
print(forecast_df)
```



```
      Date  Forecasted Branches
0  2024-11-22        33.302323
1  2024-11-23        13.771657
2  2024-11-24        45.582651
3  2024-11-25        78.298471
4  2024-11-26        89.637294
..    ...
360 2025-11-17        54.036626
361 2025-11-18        54.036625
362 2025-11-19        54.036625
363 2025-11-20        54.036626
364 2025-11-21        54.036626
```

[365 rows x 2 columns]

#Add your code for requirement 10.9 in this cell

#Add your code for requirement 10.8 in this cell

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from datetime import datetime
```

Load the data

```

df_branches = pd.read_csv("github_all_data.csv")

# Extract branch creation date (assuming the 'branch_created_at' column contains branch timestamp)
df_branches['created_at'] = pd.to_datetime(df_branches['created_at'])

# Filter branches from November 2022 to November 2024
start_date = '2022-11-01'
end_date = '2024-11-30'
df_branches = df_branches[(df_branches['created_at'] >= start_date) &
                           (df_branches['created_at'] <= end_date)]

# Group by date and count the number of branches created per day
df_branches['date'] = df_branches['created_at'].dt.date
df_branches_daily = df_branches.groupby('date').size().reset_index(name='branch_count')

# Convert 'date' to datetime and set as index
df_branches_daily['date'] = pd.to_datetime(df_branches_daily['date'])
df_branches_daily.set_index('date', inplace=True)

# Plot the original data
plt.figure(figsize=(10, 6))
plt.plot(df_branches_daily.index, df_branches_daily['branch_count'],
         label='Contributors Count')
plt.xlabel('Date')
plt.ylabel('Number of Contributors')
plt.title('Contributors Over Time (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.show()

# Fit the ARIMA model
# The (p, d, q) parameters can be tuned based on the dataset's characteristics.
model = ARIMA(df_branches_daily['branch_count'], order=(5, 1, 2))
fitted_model = model.fit()

# Forecast branches from November 2024 to November 2025
forecast_steps = 365 # Forecast for one year (adjust as needed)
forecast = fitted_model.get_forecast(steps=forecast_steps)

# Extract forecasted values and confidence intervals
forecast_index = pd.date_range(start=df_branches_daily.index[-1] +
                                pd.Timedelta(days=1), periods=forecast_steps, freq='D')
forecast_values = forecast.predicted_mean
forecast_ci = forecast.conf_int()

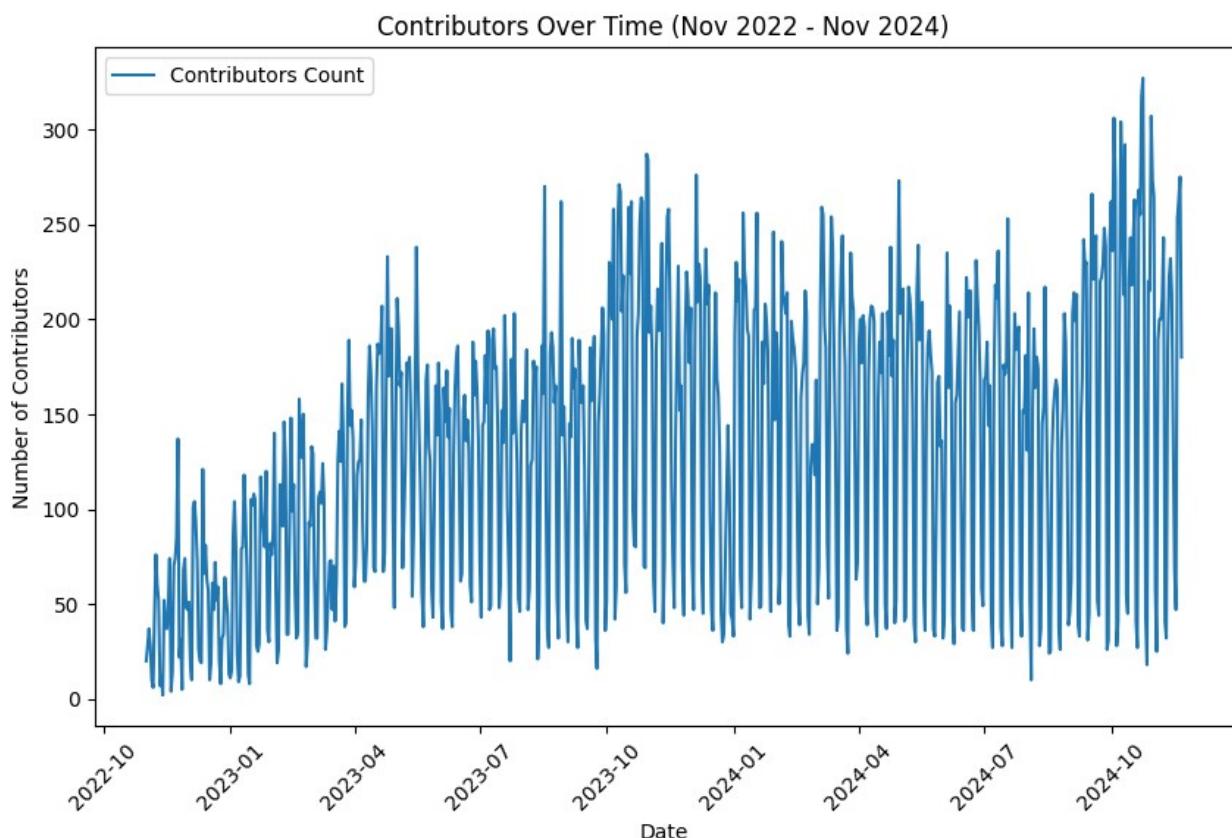
# Plot the forecast
plt.figure(figsize=(12, 6))

```

```

plt.plot(df_branches_daily.index, df_branches_daily['branch_count'],
label='Actual Contributors', color='blue')
plt.plot(forecast_index, forecast_values, label='Forecasted
Contributors', color='orange', linestyle='--')
# plt.fill_between(forecast_index, forecast_ci.iloc[:, 0],
forecast_ci.iloc[:, 1], color='orange', alpha=0.2, label='Confidence
Interval')
plt.xlabel('Date')
plt.ylabel('Number of Contributors')
plt.title('Contributors Forecast (Nov 2024 - Nov 2025)')
plt.xticks(rotation=45)
plt.legend()
plt.savefig("forecasting/StatsModel/contributors_forecast.png")
plt.show()

```



```

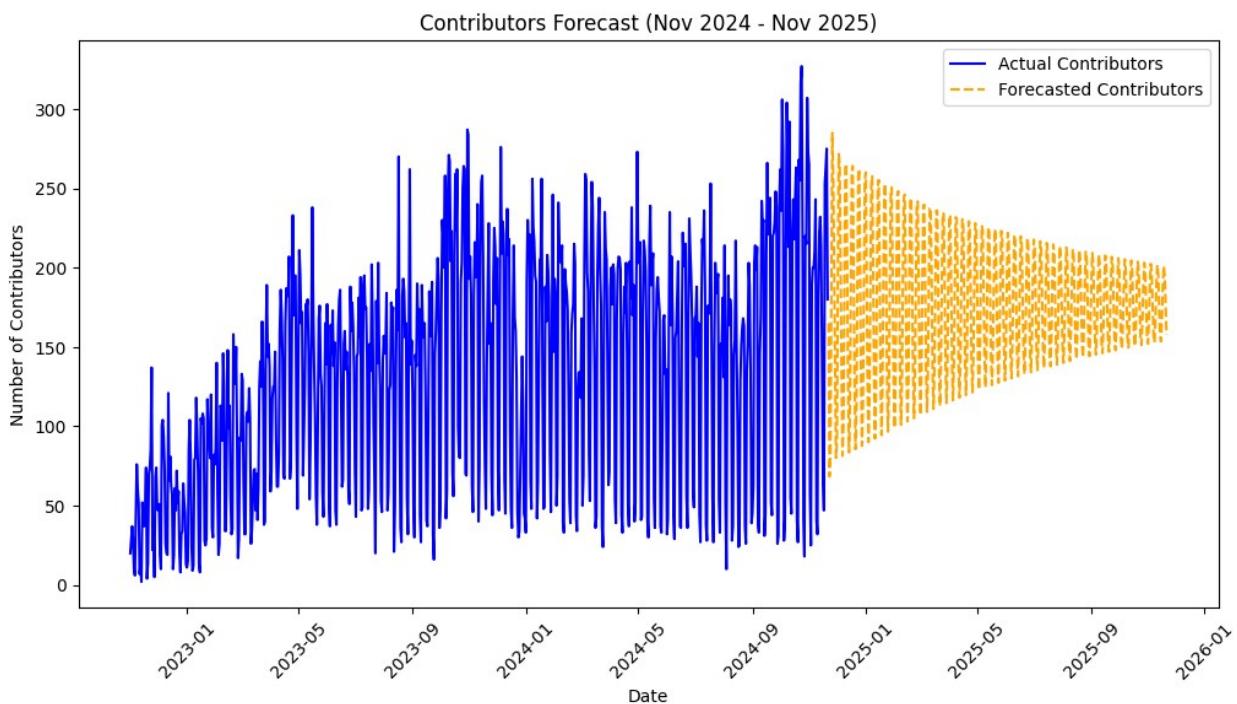
c:\Users\soham\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: No
frequency information was provided, so inferred frequency D will be
used.
    self._init_dates(dates, freq)
c:\Users\soham\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: No
frequency information was provided, so inferred frequency D will be

```

```

used.
    self._init_dates(dates, freq)
c:\Users\soham\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: No
frequency information was provided, so inferred frequency D will be
used.
    self._init_dates(dates, freq)
c:\Users\soham\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\base\model.py:607: ConvergenceWarning: Maximum
Likelihood optimization failed to converge. Check mle_retvals
warnings.warn("Maximum Likelihood optimization failed to "

```



#Add your code for requirement 10.10 in this cell

```

import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# Load the release data
df_releases = pd.read_csv("github_all_data.csv")

# Convert the 'published_at' field to datetime
df_releases['published_at'] =
pd.to_datetime(df_releases['published_at'])

# Filter data between November 2022 and November 2024
start_date = '2022-11-01'

```

```

end_date = '2024-11-30'
df_releases = df_releases[(df_releases['published_at'] >= start_date) & (df_releases['published_at'] <= end_date)]

# Group by date and count the number of releases
df_releases['date'] = df_releases['published_at'].dt.date
df_releases_daily = df_releases.groupby('date').size().reset_index(name='release_count')

# Convert 'date' to datetime and set it as the index
df_releases_daily['date'] = pd.to_datetime(df_releases_daily['date'])
df_releases_daily.set_index('date', inplace=True)

# Plot the original data
plt.figure(figsize=(10, 6))
plt.plot(df_releases_daily.index, df_releases_daily['release_count'],
label='Release Count')
plt.xlabel('Date')
plt.ylabel('Number of Releases')
plt.title('Releases Over Time (Nov 2022 - Nov 2024)')
plt.xticks(rotation=45)
plt.legend()
plt.show()

# Fit an ARIMA model
# Parameters (p, d, q) can be tuned based on ACF/PACF plots
model = ARIMA(df_releases_daily['release_count'], order=(5, 1, 2))
fitted_model = model.fit()

# Forecast releases for the next 12 months (365 days)
forecast_steps = 365
forecast = fitted_model.get_forecast(steps=forecast_steps)

# Get the forecast values and confidence intervals
forecast_index = pd.date_range(start=df_releases_daily.index[-1] + pd.Timedelta(days=1), periods=forecast_steps, freq='D')
forecast_values = forecast.predicted_mean
forecast_ci = forecast.conf_int()

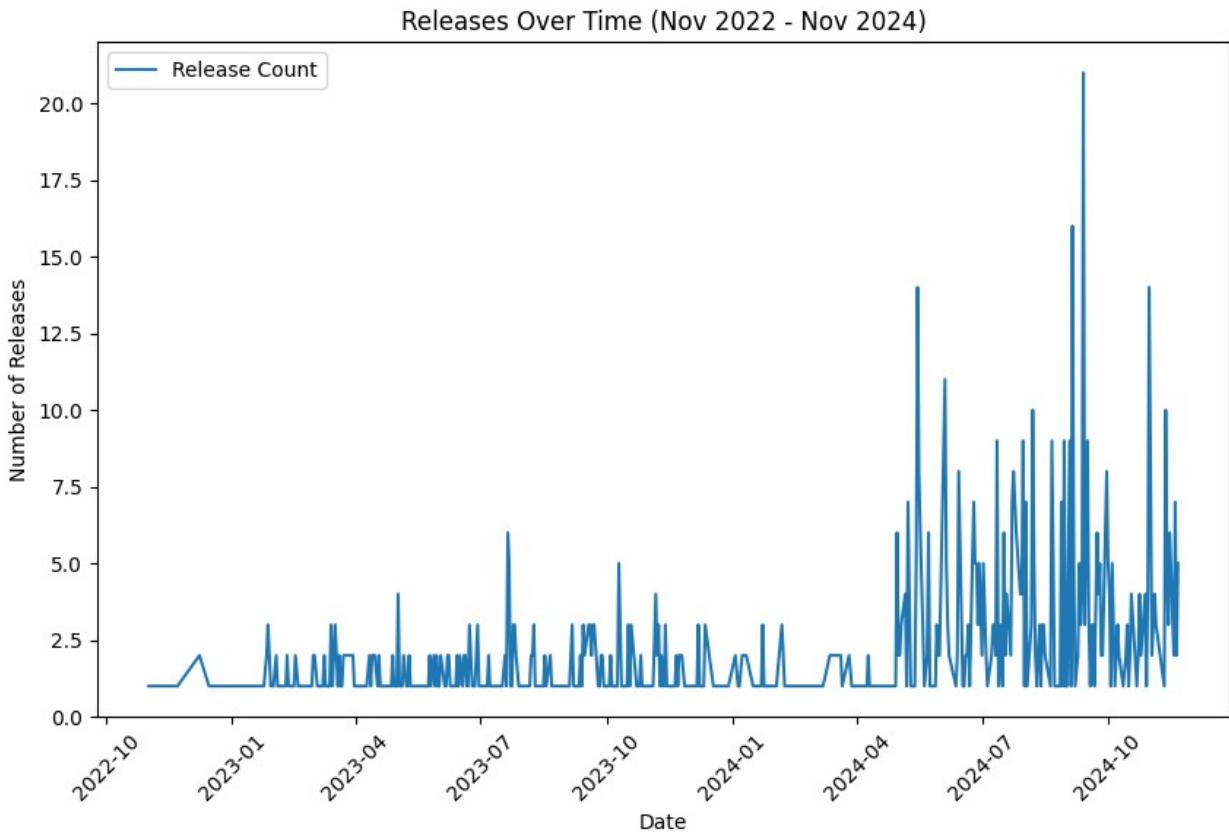
# Plot the forecast
plt.figure(figsize=(12, 6))
plt.plot(df_releases_daily.index, df_releases_daily['release_count'],
label='Actual Releases', color='blue')
plt.plot(forecast_index, forecast_values, label='Forecasted Releases',
color='orange', linestyle='--')
plt.fill_between(forecast_index, forecast_ci.iloc[:, 0],
forecast_ci.iloc[:, 1], color='orange', alpha=0.2, label='Confidence Interval')
plt.xlabel('Date')
plt.ylabel('Number of Releases')

```

```

plt.title('Releases Forecast (Nov 2024 - Nov 2025)')
plt.xticks(rotation=45)
plt.legend()
plt.savefig("forecasting/StatsModel/releases_forecast.png")
plt.show()

```

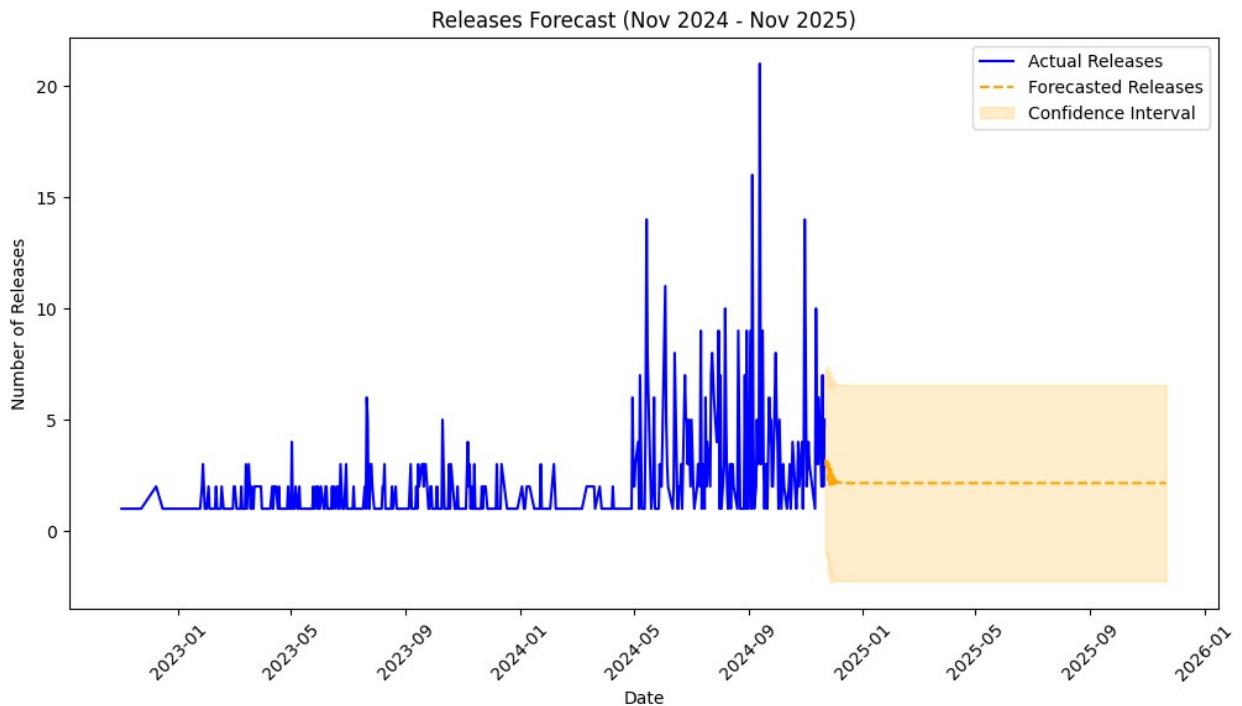


```

c:\Users\soham\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: A date
index has been provided, but it has no associated frequency
information and so will be ignored when e.g. forecasting.
    self._init_dates(dates, freq)
c:\Users\soham\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: A date
index has been provided, but it has no associated frequency
information and so will be ignored when e.g. forecasting.
    self._init_dates(dates, freq)
c:\Users\soham\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: A date
index has been provided, but it has no associated frequency
information and so will be ignored when e.g. forecasting.
    self._init_dates(dates, freq)
c:\Users\soham\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:837: ValueWarning: No

```

```
supported index is available. Prediction results will be given with an
integer index beginning at `start`.
return get_prediction_index()
```



analytics

November 22, 2024

##

Part II

##

ElasticSearch index and vector embeddings

0.1 REQUIREMENT 1

Create a Bar Chart to plot the number of issues created for every repository for every day of the week; that is total number of issues created on Monday, Tuesday, Wednesday ..., Sunday for EVERY Repo name.

```
[ ]: """
Name: Soham Sonar
CWID:A20541266
"""

import pandas as pd
import matplotlib.pyplot as plt

# Load the data
df_detailed = pd.read_csv("github_all_data.csv")

# Filter for the issues created
df_issues = df_detailed[df_detailed['type'] == 'issue']

# Convert 'created_at' to datetime format
df_issues['created_at'] = pd.to_datetime(df_issues['created_at'])

# Extract the day of the week from the 'created_at' column
df_issues['day_of_week'] = df_issues['created_at'].dt.day_name()

# Group by repository and day of the week, then count the number of issues
issue_counts = df_issues.groupby(['repository', 'day_of_week']).size()
    .unstack(fill_value=0)

# Reorder the days of the week
```

```

days_of_week_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
issue_counts = issue_counts[days_of_week_order]

# Plot the bar chart
plt.figure(figsize=(12, 6))
issue_counts.plot(kind='bar', stacked=False, figsize=(12, 6))

# Customize the plot
plt.title('Total Number of Issues Created by Day of the Week for Each Repository')
plt.xlabel('Repository')
plt.ylabel('Number of Issues Created')
plt.xticks(rotation=45)
plt.legend(title='Day of Week')
plt.tight_layout()

# Show the plot
plt.show()

```

```

C:\Users\soham\AppData\Local\Temp\ipykernel_7176\511964181.py:9: DtypeWarning:
Columns (5,6,7,8,9,10,12) have mixed types. Specify dtype option on import or
set low_memory=False.

df_detailed = pd.read_csv("github_all_data.csv")
C:\Users\soham\AppData\Local\Temp\ipykernel_7176\511964181.py:15:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

```

```

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
    df_issues['created_at'] = pd.to_datetime(df_issues['created_at'])
C:\Users\soham\AppData\Local\Temp\ipykernel_7176\511964181.py:18:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

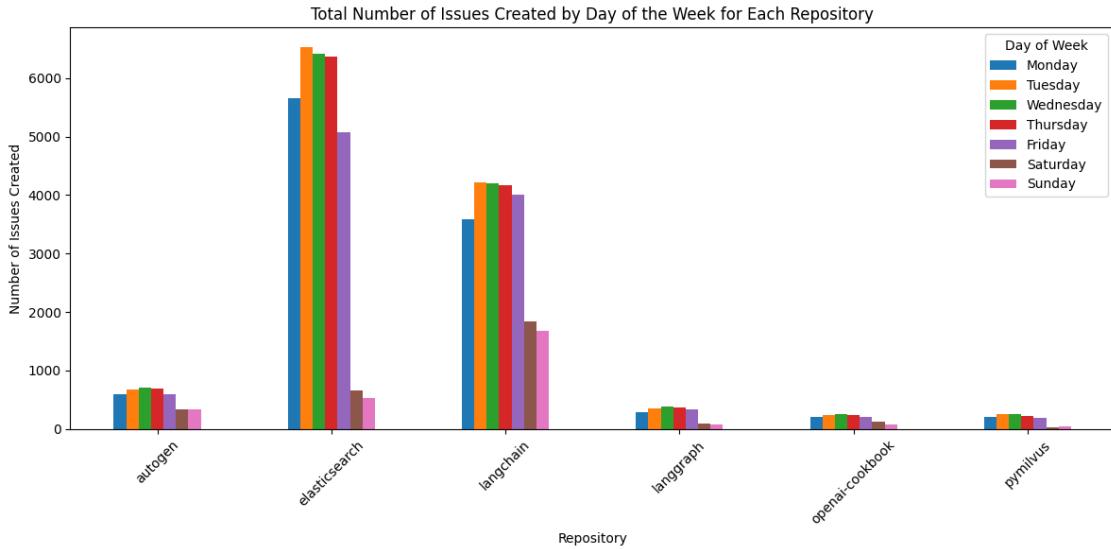
```

```

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
    df_issues['day_of_week'] = df_issues['created_at'].dt.day_name()

```

<Figure size 1200x600 with 0 Axes>



0.2 REQUIREMENT 2

Use vector embeddings, score, and semantic search to identify and list the Top 5 most similar issues for every repo listed above.

```
[10]: import requests
from datetime import datetime, timedelta
from elasticsearch import Elasticsearch

es = Elasticsearch(['http://localhost:9200'])
es.ping()    #connection testing
```

[10]: True

```
[11]: # To search documents

#Import Elasticsearch and helpers from elasticsearch

from elasticsearch import Elasticsearch,helpers
import json

es = Elasticsearch(['http://localhost:9200'])
es.ping()    #connection testing
res = es.search(index="github_issues", body={"query": {"match_all": {}}})

# Extract the total hit count
print("Got %d Hits:" % res['hits']['total']['value'])

# Extract hits from the response
```

```

hits = res.get('hits', {})

# Pretty print the hits
print(json.dumps(hits, indent=2))

```

Got 3911 Hits:

```

{
  "total": {
    "value": 3911,
    "relation": "eq"
  },
  "max_score": 1.0,
  "hits": [
    {
      "_index": "github_issues",
      "_id": "AmLQUZMBKaR7I-arfsTA",
      "_score": 1.0,
      "_source": {
        "_type": "issue",
        "_repo": "langchain",
        "_issueNumber": "28276",
        "_title": "langchain-chroma== 0.1.4 method get_by_ids is listed in documentation BUT I am getting NotImplementedError",
        "_createdAt": "2024-11-22T01:13:50Z",
        "_closedAt": "2024-12-31T00:36:30Z",
        "_state": "open",
        "_body": "### Checked other resources\n\n- [X] I added a very descriptive title to this issue.\n- [X] I searched the LangChain documentation with the integrated search.\n- [X] I used the GitHub search to find a similar question and didn't find it.\n- [X] I am sure that this is a bug in LangChain rather than my code.\n- [X] The bug is not resolved by updating to the latest stable version of LangChain (or the specific integration package).\n\n### Example Code\n\n```\r\n# HuggingFace embedding (no issue)\r\nfrom langchain_huggingface import HuggingFaceEmbeddings\r\nembeddings = HuggingFaceEmbeddings(model=\"sentence-transformers/all-mpnet-base-v2\")\r\n\r\n# create langchain-chroma persistent client with collection name 'example_collection' (no issue)\r\nfrom langchain_chroma import Chroma\r\nvector_store = Chroma(\r\n    collection_name=\"example_collection\", # collection is \"table\" in vector store\r\n    embedding_function=hf, # hf is huggingface embeddings derived from the previous step\r\n    persist_directory=\"./vectorstore/chroma_langchain_db\", # Where to save data locally, remove if not necessary\r\n)\r\n\r\n# add at least one document into vector collection (no issue)\r\nfrom uuid import uuid4\r\nfrom langchain_core.documents import Document\r\ndocument_1 = Document(\r\n    page_content=\"I had chocolate chip pancakes and scrambled eggs for breakfast this morning.\",\r\n    metadata={\"source\": \"tweet\"},\r\n)

```

```

id=1,\r\n)\r\n\r\nr\ndocuments = [\r\n    document_1,\r\n]\r\nr\nr\nuids =
[str(uuid4()) for _ in
range(len(documents))]\r\nvector_store.add_documents(documents=documents,
ids=uids)\r\n\r\n#----- ERROR ENCONTERED when running
get_by_ids \r\n# attempt to run get_by_ids yields
NotImplementedError\r\nvector_store.get_by_ids(['6314982d-455f-47cc-
bf97-6e5324f6af62'])\r\n```\n### Error Message and Stack Trace (if
applicable)\n{\r\n    "name": "NotImplementedError",\r\n    "message": "
"Chroma does not yet support get_by_ids.",\r\n    "stack": "
-----\r\nNotImplementedErro
r
Traceback (most recent call last)\r\nCell In[87], line
3\r\n      1 # testing get the first two document ids\r\n         2 # ids =
['db1e5f74-f18d-4765-a193-d30eaed7552f',
'12861b34-df54-4e40-8e1e-ae9ea901d378']\r\n----> 3
vector_store.get_by_ids(['6314982d-455f-47cc-bf97-6e5324f6af62'])\r\n      5 #
get_by_ids() functionality is not available until v0.2.11\r\n\r\nFile
~/Documents/0_-_Python_Projects/05_Gen_AI/venv_3_11/lib/python3.11/site-
packages/langchain_core/vectorstores/base.py:164, in
VectorStore.get_by_ids(self, ids)\r\n    140 \\\"\\\"\\\"\\\"Get documents by their
IDs.\r\n    141 \r\n    142 The returned documents are expected to have the ID
field set to the ID of the\r\n    (...) \r\n    161 .. versionadded:: 0.2.11\r\n
162 \\\"\\\"\\\"\\\"\\\" \r\n    163 msg = f\\\"{self.__class__.__name__} does not yet
support get_by_ids.\\\"\\\"\r\n--> 164 raise
NotImplementedError(msg)\r\n\r\nNotImplementedError: Chroma does not yet support
get_by_ids.\r\nI am just trying to run the
vector_store method `get_by_ids` - it is listed as one of the available methods
in [here](https://python.langchain.com/api_reference/chroma/vectorstores/langcha
in_chroma.vectorstores.Chroma.html)\r\n\r\nSystem Info\r\n$ python -m
langchain_core.sys_info\r\nSystem Information\r\n-----\r\nOS:
Darwin\r\n> OS Version: Darwin Kernel Version 23.6.0: Mon Jul 29 21:13:00 PDT
2024; root:xnu-10063.141.2~1/RELEASE_X86_64\r\n> Python Version: 3.11.10 (main,
Nov 19 2024, 15:24:32) [Clang 12.0.0 (clang-1200.0.32.29)]\r\n\r\nPackage
Information\r\n-----\r\nlangchain_core: 0.3.19\r\nlangchain:
0.3.7\r\nlangchain_community: 0.3.4\r\nlangsmith: 0.1.143\r\n
langchain_chroma: 0.1.4\r\nlangchain_experimental: 0.3.3\r\nlangchain_groq:
0.2.1\r\nlangchain_huggingface: 0.1.2\r\nlangchain_text_splitters:
0.3.2\r\nOptional packages not
installed\r\n-----\r\nlanggraph\r\n
langserve\r\nOther Dependencies\r\n-----\r\naiohttp:
3.11.6\r\nasync-timeout: Installed. No version info available.\r\nchromadb:
0.5.20\r\ndataclasses-json: 0.6.7\r\nfastapi: 0.115.5\r\ngroq: 0.12.0\r\n
httpx: 0.27.2\r\nhttpx-sse: 0.4.0\r\nhuggingface-hub: 0.26.2\r\njsonpatch:
1.33\r\nnumpy: 1.26.4\r\norjson: 3.10.11\r\npackaging: 24.2\r\npydantic:
2.9.2\r\npydantic-settings: 2.6.1\r\nPyYAML: 6.0.2\r\nrequests:
2.32.3\r\nrequests-toolbelt: 1.0.0\r\nsentence-transformers: 3.3.1\r\n
SQLAlchemy: 2.0.36\r\ntenacity: 9.0.0\r\ntokenizers: 0.20.3\r\n
transformers: 4.46.3\r\ntyping-extensions: 4.12.2",
"GitHub_Issue_vector": [

```

-0.02166162058711052,
-0.020060794427990913,
-0.00046240061055868864,
-0.01864745281636715,
-0.018488813191652298,
0.013455587439239025,
0.008487259037792683,
-0.02386816404759884,
-0.014645390212535858,
-0.018719563260674477,
0.01428484357893467,
-0.011977347545325756,
-0.026709267869591713,
0.030112825334072113,
-0.0004198110837023705,
0.00597425177693367,
0.02169046364724636,
-0.028540844097733498,
0.02496422454714775,
-0.0003328292805235833,
0.0053649283945560455,
0.014746342785656452,
-0.007045074366033077,
-0.008162767626345158,
-0.02771879732608795,
0.02244039997458458,
0.029449420049786568,
-0.043236710131168365,
-0.011176934465765953,
0.006338403560221195,
0.03152616694569588,
-0.007427253294736147,
-0.027992812916636467,
-0.027733219787478447,
-0.02483442798256874,
-0.023204758763313293,
0.03034357540309429,
-0.02376721054315567,
0.026464097201824188,
-0.012554221786558628,
0.005938197020441294,
0.002276849700137973,
0.009503999724984169,
-0.04666911065578461,
-0.011530269868671894,
0.04415970668196678,
-0.005779556930065155,
-0.005689420271664858,

-0.02477673999965191,
0.01293640024960041,
0.020897261798381805,
0.04614992439746857,
0.003529747948050499,
-0.01566212996840477,
0.005181049928069115,
-0.017262956127524376,
0.02704097144305706,
0.02636314369738102,
-0.0004594711645040661,
-0.009496788494288921,
0.025598784908652306,
0.003470257855951786,
-0.015849614515900612,
-0.015734240412712097,
-0.019181061536073685,
-0.0040453290566802025,
0.0015133930137380958,
0.011068770661950111,
0.02626219019293785,
0.008559368550777435,
0.01129952073097229,
-0.006900855805724859,
-0.007081128656864166,
0.0003427443152759224,
0.03478550538420677,
-0.027863016352057457,
-0.0019145007245242596,
-0.02771879732608795,
0.033516380935907364,
-0.010311623103916645,
0.03267991542816162,
-0.014393007382750511,
-0.025411302223801613,
0.027257299050688744,
0.04640951752662659,
0.03417978808283806,
0.01776771992444992,
0.022743258625268936,
0.013030142523348331,
-0.013917086645960808,
0.006630445830523968,
0.006414118222892284,
0.023392243310809135,
0.0008499878458678722,
-0.004971933085471392,
0.012684018351137638,

-0.004276078660041094,
0.018474390730261803,
0.007946440018713474,
-0.011176934465765953,
-0.02714192308485508,
0.0194550771266222,
-0.029492685571312904,
-0.011191355995833874,
-0.01306619681417942,
-0.02000310830771923,
0.014984303154051304,
-0.0021001819986850023,
0.02976670116186142,
0.011212988756597042,
-0.024632520973682404,
0.02973785623908043,
-0.018171532079577446,
-0.03008398227393627,
0.005833638831973076,
-0.03787178173661232,
-0.01646975427865982,
-0.010520740412175655,
-0.006493438500910997,
-0.007391198538243771,
0.03585272282361984,
0.020161747932434082,
0.007301062345504761,
-0.008076236583292484,
0.04672679677605629,
0.010773122310638428,
-0.0006160834454931319,
-0.03492972254753113,
-0.013917086645960808,
-0.014443484134972095,
0.021964479237794876,
-0.0030376024078577757,
0.01948392018675804,
-0.000835566024761647,
-0.010765912011265755,
0.0280505008995533,
-0.014782397076487541,
-0.012748916633427143,
-0.008761274628341198,
-0.010888497345149517,
0.016556285321712494,
0.042198337614536285,
0.0009383216965943575,
-0.019007999449968338,

-0.018012892454862595,
0.022613462060689926,
-0.009020867757499218,
0.006843168288469315,
-0.011292309500277042,
-0.010780333541333675,
0.01608036458492279,
-0.007091945502907038,
-0.01301572099328041,
-0.007477729581296444,
0.023291289806365967,
0.007816643454134464,
-0.008220454677939415,
-0.0033638968598097563,
-0.009020867757499218,
-0.009410257451236248,
-0.010109717957675457,
0.011746597476303577,
-0.009655429050326347,
-0.01890704594552517,
0.001795520423911512,
0.028252406045794487,
0.031439635902643204,
-0.0199454203248024,
-0.037727560847997665,
0.00899923499673605,
-0.0212433859705925,
-0.008321408182382584,
-0.019512765109539032,
0.016657238826155663,
0.02026270143687725,
0.016873566433787346,
0.014652600511908531,
-0.005289213731884956,
-0.02389700710773468,
-0.008674743585288525,
-0.004726761486381292,
0.028540844097733498,
0.021676043048501015,
0.027156345546245575,
-0.022411556914448738,
0.010044819675385952,
-0.010520740412175655,
-0.007308273110538721,
0.011724964715540409,
0.007463308051228523,
0.026190081611275673,
0.017839830368757248,

0.01415504701435566,
-0.0045573050156235695,
-0.5657980442047119,
-0.02195005677640438,
-0.01538811530917883,
-0.015070834197103977,
0.008328619413077831,
0.00901365652680397,
-0.02327686734497547,
0.012020613066852093,
-0.02794954739511013,
0.04707292094826698,
-0.026738112792372704,
-0.018330173566937447,
0.011450950056314468,
-0.006839562673121691,
-0.002857329323887825,
-0.02337782084941864,
0.02360857091844082,
-0.0226999931037426,
-0.0038434232119470835,
0.024848848581314087,
-0.02869948372244835,
0.025411302223801613,
0.005685814656317234,
0.0008873945334926248,
-0.01641206629574299,
0.013080619275569916,
0.010311623103916645,
-0.010787544772028923,
0.02587280049920082,
-0.005548807326704264,
-0.017104314640164375,
0.024531569331884384,
0.036487285047769547,
-0.008162767626345158,
0.043784741312265396,
0.002648212481290102,
-0.015330427326261997,
0.05936034023761749,
-0.007968072779476643,
-0.007903174497187138,
-0.018229220062494278,
-0.022296180948615074,
0.0303147304803133,
0.011083192192018032,
-0.0017711835680529475,
0.005999490153044462,

-0.002154263900592923,
0.016815878450870514,
0.0015846008900552988,
-0.014868928119540215,
0.02137318253517151,
-0.011984557844698429,
0.005761529318988323,
0.013657492585480213,
0.02275768108665943,
-0.020305966958403587,
0.006540309637784958,
-0.022916320711374283,
0.03342984989285469,
-0.011515848338603973,
-0.007989705540239811,
0.007156843785196543,
-0.008061815053224564,
0.0014791410649195313,
-0.01825806312263012,
0.026190081611275673,
0.010044819675385952,
0.001874840585514903,
0.007283034734427929,
-0.03368944302201271,
0.01841670460999012,
-0.000558396044652909,
-0.03184344619512558,
0.0047664218582212925,
-0.009907811880111694,
0.01706104911863804,
0.03279528766870499,
0.00720732007175684,
-0.013967562466859818,
0.009778015315532684,
0.017580237239599228,
-0.008350252173841,
-0.042659834027290344,
0.00426526227965951,
0.023550882935523987,
0.00801133830100298,
-0.029463842511177063,
-0.0036288981791585684,
-0.013578172773122787,
-0.035881564021110535,
0.003120528068393469,
0.03706415742635727,
0.0002726631355471909,
-0.03951587155461311,

0.01628226973116398,
0.01378007885068655,
0.01291476748837242,
-0.026017019525170326,
-0.0006178861949592829,
-0.039544716477394104,
-0.019498342648148537,
0.007607526611536741,
-0.014630967751145363,
0.008696376346051693,
0.02275768108665943,
-0.016008254140615463,
-0.026694847270846367,
0.004038118291646242,
0.02490653656423092,
0.010744279250502586,
-0.022714415565133095,
-0.00055524124763906,
-0.002707702573388815,
0.021200120449066162,
0.0032214808743447065,
-0.03836212307214737,
-0.005411799531430006,
-0.0045284610241651535,
0.005858877208083868,
-0.006388879846781492,
0.01877724938094616,
-0.019382968544960022,
-0.02192121371626854,
-0.0026013413444161415,
0.035650815814733505,
-0.012395581230521202,
-0.022555775940418243,
-0.011068770661950111,
-0.014493959955871105,
0.020738622173666954,
0.006349219940602779,
-0.00959774199873209,
-0.004705128725618124,
-0.017984047532081604,
-0.017464861273765564,
0.04029465094208717,
0.02033481001853943,
0.00610404834151268,
0.02389700710773468,
-0.01929643750190735,
0.015359271317720413,
-0.007257796358317137,

0.01551791187375784,
0.0021957268472760916,
-0.02649294026196003,
-0.021228965371847153,
-0.007344327867031097,
-0.009064133279025555,
-0.005519963335245848,
-0.00909297727048397,
-0.024098912253975868,
-0.018402282148599625,
-0.036487285047769547,
-0.006944121327251196,
0.012705651111900806,
-0.00251661310903728,
-0.008422360755503178,
-0.037727560847997665,
0.003893899731338024,
0.005094518885016441,
0.008775696158409119,
0.015489067882299423,
-0.011176934465765953,
0.009193929843604565,
-0.01903684251010418,
-0.013679125346243382,
0.007470518816262484,
0.018474390730261803,
-0.030978135764598846,
-0.02558436430990696,
-0.011162512935698032,
-0.01867629773914814,
0.009460734203457832,
0.007564261090010405,
0.01693125255405903,
0.02246924489736557,
-0.007127999793738127,
0.0030195750296115875,
-0.028483156114816666,
-0.0046222032979130745,
-0.0045428830198943615,
0.0057182637974619865,
0.0004231912025716156,
-0.022296180948615074,
0.028137031942605972,
0.015186209231615067,
0.011638433672487736,
0.013044564053416252,
0.0031043034978210926,
0.024820005521178246,

-0.002098379423841834,
-0.006929699331521988,
-0.02616123855113983,
0.0065691531635820866,
0.006453778129070997,
0.0028014446143060923,
0.011407684534788132,
0.01838785968720913,
-0.014371374621987343,
0.02143087051808834,
-0.0011474385391920805,
0.02207985334098339,
-0.004852952901273966,
-0.014868928119540215,
0.00022365136828739196,
-0.02062324620783329,
0.00888385996222496,
-0.04049655795097351,
0.0038578452076762915,
-0.008761274628341198,
0.01319599337875843,
0.022267337888479233,
-0.02078188769519329,
-0.012756126932799816,
-0.014782397076487541,
-0.01357096154242754,
-0.010729856789112091,
0.008811751380562782,
-0.007470518816262484,
0.013008509762585163,
-0.026074707508087158,
0.02724287658929825,
0.0020424947142601013,
-0.0010708224726840854,
-0.002143447520211339,
-0.022224072366952896,
0.020926106721162796,
-0.01740717515349388,
0.0212433859705925,
-0.007448886055499315,
-0.01851765625178814,
0.021286651492118835,
0.0042688678950071335,
-0.0025364430621266365,
0.019729092717170715,
0.009006446227431297,
0.06755194813013077,
-0.016123630106449127,

0.014840085059404373,
-0.030430106446146965,
-0.007549839094281197,
0.019959842786192894,
-0.018503235653042793,
-0.0055524124763906,
0.03184344619512558,
0.0035081151872873306,
0.012373948469758034,
0.025079598650336266,
-0.011667277663946152,
0.011638433672487736,
-0.016397643834352493,
-0.002096576616168022,
-0.016253426671028137,
-0.015993833541870117,
0.005512752570211887,
-0.02184910513460636,
0.010953395627439022,
0.016253426671028137,
0.004113832954317331,
0.01838785968720913,
0.007434464059770107,
-0.019512765109539032,
0.03002629429101944,
0.008797328919172287,
0.011068770661950111,
0.0015656722243875265,
-0.0038326068315654993,
-0.03732375055551529,
-0.009273250587284565,
-0.019729092717170715,
-0.0199454203248024,
-0.022526931017637253,
-0.027444781735539436,
-0.02766111120581627,
0.023421086370944977,
0.005703842267394066,
0.016455331817269325,
0.013080619275569916,
0.03363175690174103,
0.018214797601103783,
0.0012321668909862638,
-0.02288747765123844,
0.021070323884487152,
0.004719550721347332,
0.0022750471252948046,
-0.017306221649050713,

-0.021214542910456657,
0.019700247794389725,
-0.029290780425071716,
0.001109581207856536,
-0.03233379125595093,
0.02386816404759884,
-0.003342264099046588,
-0.011991769075393677,
-0.034035567194223404,
0.017032206058502197,
0.047159451991319656,
-0.017176425084471703,
-0.004921456798911095,
0.01724853366613388,
-0.019830044358968735,
0.030718542635440826,
-0.0212433859705925,
0.018921468406915665,
0.05748549848794937,
-0.006331192795187235,
0.013722390867769718,
-0.021704886108636856,
0.008710797876119614,
-0.018359016627073288,
-0.008292564190924168,
-0.008422360755503178,
-0.009828491136431694,
-0.012568643316626549,
0.003612673608586192,
0.00775895593687892,
-0.027805328369140625,
-0.0005858876975253224,
0.02888696826994419,
-0.018791671842336655,
-0.014695866033434868,
-0.05685093626379967,
0.005999490153044462,
-0.002781614428386092,
-0.004157098475843668,
0.01909453049302101,
-0.015301584266126156,
-0.012056667357683182,
-0.016455331817269325,
0.008984813466668129,
-0.015676552429795265,
-0.022844212129712105,
-0.0030195750296115875,
0.008826172910630703,

-0.014234366826713085,
-0.03132425993680954,
0.006803508382290602,
0.0031619907822459936,
0.03847749903798103,
0.03025704436004162,
0.0004326555354055017,
-0.002439095638692379,
0.008689165115356445,
0.016253426671028137,
0.0049863550812006,
0.003911926876753569,
0.0025851167738437653,
0.02642083168029785,
0.02681022137403488,
0.011277887970209122,
0.0061004431918263435,
0.005779556930065155,
0.0030033504590392113,
-0.024488303810358047,
-0.006028333678841591,
0.0018928679637610912,
0.004023696295917034,
0.026853486895561218,
-0.010527951642870903,
0.020349232479929924,
0.04066962003707886,
-0.0004313034878578037,
0.003545972751453519,
0.007405620533972979,
0.03086276166141033,
0.025137286633253098,
-0.013282524421811104,
-0.002716716146096587,
-0.00290239742025733,
-0.007448886055499315,
-0.024358507245779037,
0.003262943821027875,
-0.009460734203457832,
-0.010657748207449913,
0.02532477118074894,
0.03441053628921509,
-0.04989960417151451,
-0.015734240412712097,
0.024603677913546562,
0.016657238826155663,
-0.013369056396186352,
-0.005368534009903669,

-0.020926106721162796,
0.0007688649347983301,
-0.0012294627958908677,
-0.03841981291770935,
-0.0063456143252551556,
0.01036209985613823,
-0.008343040943145752,
-0.0023832109291106462,
-0.012720072641968727,
0.013852188363671303,
-0.018301328644156456,
0.003947981633245945,
0.01633995771408081,
-0.00572186941280961,
-0.015964988619089127,
-0.004149887710809708,
0.02571416087448597,
-0.013592594303190708,
-0.0031006978824734688,
0.010015975683927536,
0.017522549256682396,
0.025281505659222603,
-0.025512253865599632,
-0.05537990853190422,
0.02059440314769745,
-0.027214033529162407,
0.0030015476513653994,
0.002442701021209359,
0.010614482685923576,
-0.003320631105452776,
-0.03732375055551529,
0.016887987032532692,
0.021877948194742203,
0.00249678292311728,
0.009677061811089516,
-0.012799392454326153,
0.013419532217085361,
0.02763226628303528,
0.00762915937229991,
0.03977546468377113,
-0.014118991792201996,
-0.005855271592736244,
0.005278397351503372,
-0.02036365307867527,
-0.03426631912589073,
-0.012849869206547737,
-0.03518931567668915,
0.0015692777233198285,

0.009287672117352486,
-0.004344582557678223,
-0.006626840680837631,
0.009266039356589317,
0.016037099063396454,
-0.020969372242689133,
0.0028032471891492605,
-0.006590785924345255,
0.011717754416167736,
-0.003814579686149955,
-0.004438324831426144,
0.010189037770032883,
-0.008040182292461395,
-0.015445802360773087,
0.01194850355386734,
-0.016743769869208336,
0.03192997723817825,
0.02532477118074894,
-0.006601602304726839,
-0.0040741730481386185,
-0.006049966439604759,
-0.025916066020727158,
-0.014760764315724373,
0.0029366493690758944,
0.018661875277757645,
0.02947826310992241,
-0.0012691229349002242,
-0.004193153232336044,
-0.016815878450870514,
-0.02425755374133587,
-0.005963435396552086,
0.02078188769519329,
-0.00635643070563674,
-0.0061509194783866405,
0.0010203459532931447,
-0.019930997863411903,
-0.009633796289563179,
0.0008797328919172287,
-0.022022167220711708,
-0.03126657381653786,
-0.02911771647632122,
0.006803508382290602,
-0.0026319879107177258,
0.0020659302826970816,
0.008674743585288525,
-0.003825396066531539,
-0.019541608169674873,
-0.013109462335705757,

-0.02223849482834339,
0.016887987032532692,
0.0013331198133528233,
0.010347678326070309,
0.0453999862074852,
0.0337182879447937,
0.08422361314296722,
-0.0228586345911026,
-0.00639609107747674,
-0.0005872397450730205,
0.005465881433337927,
0.0030862761195749044,
-0.016065942123532295,
0.012705651111900806,
0.0032413110602647066,
-0.007308273110538721,
0.017782142385840416,
-0.005469487048685551,
-0.0062013957649469376,
-0.01365028228610754,
-0.021603932604193687,
0.00597425177693367,
-0.003185426350682974,
0.00974196009337902,
-0.02976670116186142,
-0.033574070781469345,
-0.008970391005277634,
0.02153182402253151,
0.004914245568215847,
-0.0036469255574047565,
-0.04522692412137985,
-0.013707969337701797,
0.02711308002471924,
-0.031439635902643204,
0.008090658113360405,
-0.007477729581296444,
0.034583598375320435,
-0.0026211715303361416,
0.019570451229810715,
0.0008788315462879837,
0.02818029746413231,
0.006670106202363968,
0.003610871033743024,
-0.027228454127907753,
0.0014818451600149274,
0.006587180308997631,
-0.0033693050500005484,
0.031497322022914886,

0.019310858100652695,
-0.017464861273765564,
-0.028036078438162804,
0.03253569453954697,
-0.007492151577025652,
-0.027675531804561615,
-0.013635859824717045,
0.005656971130520105,
-0.004535672254860401,
-0.036746878176927567,
-0.013397899456322193,
-0.012979665771126747,
-0.006453778129070997,
0.00840072799474001,
0.006698949728161097,
0.017623502761125565,
-0.024300819262862206,
-0.03986199572682381,
0.020608825609087944,
-0.00485655851662159,
0.030401261523365974,
0.005566834472119808,
0.05359159782528877,
0.027098657563328743,
0.0016134445322677493,
-0.017436018213629723,
-0.011811495758593082,
0.0033008011523634195,
-0.006853984668850899,
0.03865056112408638,
0.020940527319908142,
-0.02282978966832161,
-0.0011348193511366844,
0.01789751648902893,
-0.01512852218002081,
-0.03867940604686737,
-0.039054371416568756,
0.042717523872852325,
-6.574336293851957e-05,
0.02662273682653904,
-0.013123884797096252,
-0.024358507245779037,
-0.0535050667822361,
0.01245326828211546,
-0.0026103551499545574,
0.014955459162592888,
0.016138050705194473,
-0.0053649283945560455,

-0.011032716371119022,
0.010405365377664566,
-0.024848848581314087,
0.029622482135891914,
0.026694847270846367,
0.00912903156131506,
-0.009042500518262386,
-0.009936654940247536,
-0.01579192653298378,
0.02506517805159092,
0.031756915152072906,
0.029103295877575874,
-0.0053793503902852535,
-0.013729602098464966,
0.03804484382271767,
-0.016743769869208336,
-0.0160515196621418,
-0.02052229456603527,
-0.005346901249140501,
0.04225602373480797,
-0.022916320711374283,
0.012561432085931301,
-0.01488335058093071,
-0.014688655734062195,
0.011905238032341003,
0.0019775964319705963,
-0.001664822455495596,
-0.012748916633427143,
-0.0007756251725368202,
0.011003872379660606,
0.017363909631967545,
0.0009446312324143946,
0.0064898328855633736,
0.002860934706404805,
-0.01057121716439724,
-0.001392610021866858,
-0.017277376726269722,
-0.032218415290117264,
0.03207419812679291,
-0.03717953339219093,
-0.015964988619089127,
-0.03126657381653786,
-0.005801189690828323,
0.027084236964583397,
-0.024142179638147354,
0.018690718337893486,
0.007694057654589415,
0.05203403905034065,

-0.015099678188562393,
0.04066962003707886,
0.011638433672487736,
0.0023111016489565372,
-0.041333023458719254,
-0.01932528056204319,
-0.004167914856225252,
-0.009958288632333279,
0.0018270682776346803,
0.012424424290657043,
-0.03086276166141033,
0.0028807646594941616,
0.01608036458492279,
-4.549080040305853e-05,
0.025656472891569138,
-0.01812826655805111,
0.014198312535881996,
-0.012035034596920013,
0.02036365307867527,
-0.04447698965668678,
-0.004402270074933767,
0.037237219512462616,
-0.029305201023817062,
0.003116922453045845,
0.014969881623983383,
0.0036613475531339645,
0.017782142385840416,
-0.019282015040516853,
0.02327686734497547,
-0.014573280699551105,
-0.0018838542746379972,
0.010023186914622784,
-0.004503223113715649,
-0.010931762866675854,
-0.008018549531698227,
-0.0057471077889204025,
-0.026666002348065376,
-0.04144839942455292,
-0.023622991517186165,
0.02908887341618538,
0.00933814886957407,
0.007434464059770107,
-0.0017098906682804227,
0.024430615827441216,
0.032737601548433304,
0.027286142110824585,
-0.025396879762411118,
-0.017104314640164375,

-0.022022167220711708,
0.013008509762585163,
0.006352825555950403,
-0.014969881623983383,
-0.013116673566401005,
0.03233379125595093,
0.017017783597111702,
-0.015489067882299423,
-0.014371374621987343,
-0.00901365652680397,
-0.031151197850704193,
-0.00585166597738862,
-0.0005615508416667581,
0.00784548744559288,
0.003663150127977133,
-0.001878446084447205,
-0.021488558501005173,
0.006969359703361988,
-0.013715180568397045,
0.014010827988386154,
-0.028526421636343002,
0.008955969475209713,
-0.000415754911955446,
0.017623502761125565,
0.0032268890645354986,
-0.0005795781617052853,
-0.029709013178944588,
-0.006745820865035057,
0.0288581233471632,
0.01378007885068655,
0.02768995426595211,
0.027805328369140625,
-0.01909453049302101,
-0.004892612807452679,
-0.007557049859315157,
-0.020291544497013092,
-0.01773887686431408,
-0.021805839613080025,
0.01327531412243843,
-0.002170488703995943,
-0.028324514627456665,
0.01693125255405903,
-0.008162767626345158,
-0.0024625309742987156,
-0.016758190467953682,
0.007200109306722879,
-0.0027581790927797556,
0.010095295496284962,

0.006511465646326542,
-0.0019902153871953487,
0.006053572054952383,
-0.01019624900072813,
0.02379605360329151,
0.004330160561949015,
0.02130107395350933,
0.020248278975486755,
-0.0002087788307107985,
0.013981984928250313,
0.008069025352597237,
0.034064412117004395,
-0.016455331817269325,
-0.015546755865216255,
0.018863780423998833,
0.002150658518075943,
0.03141079097986221,
-0.022151963785290718,
0.023060539737343788,
-0.02732940763235092,
-0.014270422048866749,
-0.006432145368307829,
-0.0160515196621418,
-0.01219367515295744,
-0.0018036327091977,
-0.025339191779494286,
-0.014003617689013481,
-0.009424679912626743,
0.011977347545325756,
0.033401008695364,
-0.0037929469253867865,
-0.02629103511571884,
-0.019671404734253883,
-0.029060030356049538,
-0.01071543525904417,
0.008083447813987732,
-0.008710797876119614,
-0.04072730615735054,
0.0006769256433472037,
0.004669074434787035,
-0.029261935502290726,
-0.0012321668909862638,
0.2049056589603424,
-0.011148090474307537,
0.014710288494825363,
0.020709777250885963,
-0.007066707126796246,
0.029132138937711716,

-0.011277887970209122,
-0.000621491635683924,
-0.018921468406915665,
0.0038398178294301033,
0.022656727582216263,
-0.0052495538257062435,
0.012972455471754074,
-0.005584861617535353,
-0.009641007520258427,
0.03417978808283806,
-0.02882928028702736,
-0.02386816404759884,
-0.026060285046696663,
0.004438324831426144,
-0.008436783216893673,
-0.003322433913126588,
0.004539277404546738,
-0.030458949506282806,
0.016138050705194473,
-0.01968582719564438,
-0.02098379284143448,
0.009359781630337238,
-0.0023886191193014383,
0.005483909044414759,
-0.012741705402731895,
0.014912193641066551,
0.0034450197126716375,
-0.021935636177659035,
-0.008768485859036446,
-0.002648212481290102,
0.029910918325185776,
0.020839575678110123,
0.013657492585480213,
0.023594148457050323,
0.0009144354844465852,
-0.03267991542816162,
0.008422360755503178,
-0.026579471305012703,
-7.83061477704905e-05,
0.015489067882299423,
-0.005426221527159214,
0.011984557844698429,
-0.00998713169246912,
-0.0023453535977751017,
-0.017306221649050713,
-0.00221735960803926,
0.03337216377258301,
0.020406918600201607,

0.009395835921168327,
-0.036718033254146576,
-0.00712439464405179,
-0.011068770661950111,
-0.026060285046696663,
0.0036955992691218853,
-0.02020501345396042,
0.012835447676479816,
0.017753299325704575,
0.02010405994951725,
-0.022512510418891907,
0.007982494309544563,
-0.029853232204914093,
-0.017147580161690712,
-0.013289735652506351,
-0.0061364974826574326,
0.004027301911264658,
-0.013707969337701797,
0.0005921972333453596,
0.014566069468855858,
-0.01569097489118576,
-0.024329662322998047,
0.03132425993680954,
-0.009778015315532684,
0.011386051774024963,
0.04000621661543846,
0.01867629773914814,
-0.02976670116186142,
-0.011494215577840805,
0.02272883802652359,
-0.02412775717675686,
-0.01987330988049507,
0.019123375415802002,
-0.010686591267585754,
-0.010823599062860012,
-0.019411811605095863,
0.010145772248506546,
0.0015719818184152246,
-0.018301328644156456,
0.011133668944239616,
-0.007301062345504761,
0.004438324831426144,
-0.007019835989922285,
0.015734240412712097,
-0.0056173112243413925,
-0.008040182292461395,
-0.028511999174952507,
0.045803800225257874,

0.02153182402253151,
0.014854506589472294,
-0.01633995771408081,
-0.0021272229496389627,
0.002572497585788369,
-0.020536715164780617,
-0.02065209113061428,
-0.0017468467121943831,
-0.0012060273438692093,
-0.03415094316005707,
-0.008595422841608524,
-0.0003488285292405635,
0.0005998588749207556,
0.01646975427865982,
0.003771314164623618,
-0.025512253865599632,
0.03412209823727608,
0.017017783597111702,
0.013260891661047935,
-0.02023385651409626,
-0.004614992532879114,
0.014912193641066551,
0.0039047161117196083,
-0.020060794427990913,
-0.03285297751426697,
-0.00038646053872071207,
-0.011018293909728527,
-0.03530469164252281,
0.0025851167738437653,
0.016743769869208336,
0.024603677913546562,
-0.008905492722988129,
-0.005213499069213867,
-0.01006645243614912,
-0.033026039600372314,
-0.007326300255954266,
-0.006929699331521988,
-0.016109207645058632,
0.004092200193554163,
-0.0012709256261587143,
-0.005047647748142481,
-0.008523314259946346,
-0.010636115446686745,
-0.023969115689396858,
-0.012748916633427143,
0.004171520471572876,
-0.018373439088463783,
-0.03417978808283806,

-0.0028284855652600527,
-0.022714415565133095,
0.011775441467761993,
0.005779556930065155,
0.03270875662565231,
-0.023810476064682007,
-0.011039926670491695,
-0.02862737514078617,
-0.020479029044508934,
-0.005534385330975056,
-0.00776616670191288,
-0.0013682731660082936,
0.020277122035622597,
0.018459970131516457,
-0.04548651725053787,
-0.028656218200922012,
-0.1844843178987503,
0.018041735514998436,
0.036141157150268555,
-0.04750557616353035,
0.04358283430337906,
-0.0013962154043838382,
0.014580490998923779,
-0.011198567226529121,
0.016700504347682,
-0.008768485859036446,
0.027314985170960426,
-0.007968072779476643,
-0.018373439088463783,
-0.02587280049920082,
-0.020031951367855072,
0.016787035390734673,
-0.016902409493923187,
0.003394543193280697,
0.03689109534025192,
0.008826172910630703,
0.04620761051774025,
-0.034554753452539444,
0.0029781123157590628,
-0.0046871015802025795,
-0.004917851183563471,
0.0017792958533391356,
-0.005404588766396046,
0.008285353891551495,
0.0031872291583567858,
-0.016902409493923187,
-0.007268612738698721,
0.010239514522254467,

0.020579980686306953,
-0.025916066020727158,
0.035939253866672516,
0.006262688897550106,
0.03129541501402855,
-0.008948758244514465,
-0.006980176083743572,
0.023204758763313293,
0.03784293681383133,
0.015041991136968136,
0.011638433672487736,
-0.02052229456603527,
-0.0017062852857634425,
-0.01160958968102932,
0.041073430329561234,
-0.01753697171807289,
0.0024517145939171314,
-0.008617055602371693,
0.011977347545325756,
-0.02075304463505745,
-0.012799392454326153,
0.02483442798256874,
0.0029384521767497063,
0.025786269456148148,
-0.02363741397857666,
0.026348721235990524,
-0.010773122310638428,
-0.005476697813719511,
0.016815878450870514,
-0.01344116497784853,
-0.003133147256448865,
-0.014854506589472294,
-0.008797328919172287,
-0.018733983859419823,
-0.017522549256682396,
-0.0017035811906680465,
-0.012085511349141598,
0.012244151905179024,
-9.836153185460716e-05,
0.01890704594552517,
0.013080619275569916,
-0.012150409631431103,
0.02745920419692993,
0.03008398227393627,
-0.006482622120529413,
0.020291544497013092,
0.019959842786192894,
-0.00862426683306694,

-0.01857534423470497,
0.035910408943891525,
-0.010160193778574467,
0.013953140936791897,
0.03686225041747093,
0.006114864721894264,
0.008905492722988129,
-0.024733474478125572,
-0.029189826920628548,
0.0030736569315195084,
0.013239258900284767,
-0.016585128381848335,
0.019209906458854675,
-0.010174616239964962,
0.0010861456394195557,
0.0016242610290646553,
0.0226999931037426,
-0.006706160958856344,
0.013953140936791897,
-0.0012988679809495807,
0.007578682620078325,
0.008386306464672089,
-0.017594657838344574,
0.00469431234523654,
0.017277376726269722,
0.020320387557148933,
-0.033083725720644,
0.027516892179846764,
0.03582387790083885,
-0.03527584671974182,
-0.02779090777039528,
0.008667532354593277,
0.008552157320082188,
0.05021688714623451,
-0.0039047161117196083,
0.04257330298423767,
-0.010145772248506546,
-0.0025274294894188643,
0.00920835230499506,
0.019527185708284378,
0.050534166395664215,
-0.015907302498817444,
-0.025209395214915276,
0.002933043986558914,
-0.02711308002471924,
-0.01024672482162714,
-0.0794355571269989,
-0.01366470381617546,

0.0033044065348803997,
-0.014378585268259,
-0.010037608444690704,
-0.015777505934238434,
-0.011724964715540409,
0.024531569331884384,
-0.035102784633636475,
0.015979411080479622,
-0.014811241067945957,
0.007520995568484068,
-0.0033494748640805483,
0.01241721399128437,
0.016945675015449524,
0.0007760758744552732,
-0.01219367515295744,
0.009684273041784763,
-0.028122609481215477,
0.018402282148599625,
0.014385796152055264,
-0.011422106064856052,
0.008032971061766148,
0.014594913460314274,
-0.03547775372862816,
0.017493706196546555,
-0.04283289611339569,
0.014479538425803185,
-0.002011848147958517,
0.017984047532081604,
0.00712439464405179,
-0.01971467025578022,
0.0023435507901012897,
0.004081383813172579,
-0.004171520471572876,
-0.0025995387695729733,
3.256183481425978e-05,
-0.012568643316626549,
0.023594148457050323,
-0.017493706196546555,
0.014234366826713085,
0.016426488757133484,
-0.008443993516266346,
-0.012006190605461597,
-0.003459441475570202,
-0.017984047532081604,
0.021286651492118835,
0.015979411080479622,
0.0062013957649469376,
-0.024892114102840424,

-0.03640075400471687,
0.022296180948615074,
-0.0030862761195749044,
0.003911926876753569,
0.006118470337241888,
-0.0003213368763681501,
0.001037471927702427,
0.01392429694533348,
-0.018142689019441605,
-0.029175404459238052,
-0.001885657082311809,
-0.019152218475937843,
-0.004103016573935747,
0.008674743585288525,
0.015186209231615067,
-0.009439101442694664,
0.006716977339237928,
-0.014695866033434868,
0.008718009106814861,
-0.024401772767305374,
-0.00041868435801006854,
0.023983538150787354,
-0.005318057723343372,
0.000311421841615811,
0.0011438330402597785,
-0.01181870698928833,
0.009705905802547932,
-0.013116673566401005,
0.010520740412175655,
-0.013419532217085361,
-0.009395835921168327,
0.010967818088829517,
-0.007088339887559414,
0.012489323504269123,
0.0042688678950071335,
0.012049456126987934,
0.02428639680147171,
0.028165875002741814,
-0.0028717510867863894,
-0.057312436401844025,
0.003268352011218667,
0.016541862860322,
0.026507362723350525,
-0.010816388763487339,
0.0027491652872413397,
0.024300819262862206,
-0.019830044358968735,
0.002298482460901141,

0.014566069468855858,
-0.0011879999656230211,
-0.06460989266633987,
-0.011696121655404568,
-0.06114864721894264,
0.03787178173661232,
0.0062951380386948586,
-0.036487285047769547,
-0.01196292508393526,
-0.05163022503256798,
0.0035820272751152515,
-0.017032206058502197,
0.026247769594192505,
0.022252915427088737,
0.003872266970574856,
-0.010434209369122982,
-0.018070578575134277,
-0.009655429050326347,
0.00776616670191288,
0.012597487308084965,
0.023550882935523987,
-0.006392485462129116,
0.036487285047769547,
0.011162512935698032,
-0.006529493257403374,
0.0024156600702553988,
0.00860984530299902,
0.015864036977291107,
-0.0031800181604921818,
0.02117127738893032,
0.0005439742235466838,
0.0415060892701149,
0.003142160829156637,
-0.02327686734497547,
0.029651325196027756,
-0.019397389143705368,
-0.0006701654056087136,
0.023695101961493492,
0.024012381210923195,
-0.008732430636882782,
0.012352315708994865,
0.010650536976754665,
0.04144839942455292,
-0.004131860565394163,
-0.026190081611275673,
-0.03830443695187569,
0.01292197871953249,
-0.01344116497784853,

0.004077778197824955,
0.016844721511006355,
0.004726761486381292,
0.016383223235607147,
0.00626629451289773,
0.02137318253517151,
0.026507362723350525,
0.008818961679935455,
-0.014082937501370907,
-0.028612952679395676,
0.0016936660977080464,
-0.04467889294028282,
-0.0013935113092884421,
0.014796819537878036,
-0.0001355428685201332,
0.02587280049920082,
0.04092921316623688,
0.004279684275388718,
0.038015998899936676,
-0.02192121371626854,
0.024820005521178246,
-0.014630967751145363,
-0.03311257064342499,
-0.01424878928810358,
-0.007012625224888325,
-0.011328363791108131,
-0.017262956127524376,
-0.012979665771126747,
0.005656971130520105,
0.01301572099328041,
-0.007113578263670206,
0.004878191277384758,
-0.013592594303190708,
-0.011465371586382389,
0.017436018213629723,
0.025642050430178642,
0.020695356652140617,
-0.02143087051808834,
-0.03862171620130539,
0.004510433878749609,
0.03083391673862934,
0.02153182402253151,
-0.023680679500102997,
0.011898026801645756,
0.013188783079385757,
-0.028656218200922012,
-0.031468480825424194,
0.03963124752044678,

0.01760908029973507,
0.006951332092285156,
-0.016541862860322,
0.020378075540065765,
-0.0023038906510919333,
0.01589288003742695,
0.02862737514078617,
-0.007679635658860207,
0.006742215249687433,
-0.007730112411081791,
0.0023038906510919333,
-0.01476797554641962,
-0.016138050705194473,
0.020637668669223785,
-0.012748916633427143,
-0.013383477926254272,
0.010693802498281002,
-0.007055890746414661,
-0.022916320711374283,
-0.007275823969393969,
-0.014479538425803185,
0.029319623485207558,
-0.009532843716442585,
-0.0055091469548642635,
-0.011306731030344963,
-0.010628904215991497,
-0.03634306415915489,
0.03368944302201271,
0.013073408044874668,
0.00751378433778882,
0.024978647008538246,
-0.022873055189847946,
0.026925595477223396,
0.03816021978855133,
0.0021001819986850023,
0.0046726795844733715,
0.03582387790083885,
0.00762915937229991,
-0.00639609107747674,
0.011616800911724567,
-0.024171022698283195,
-0.016642816364765167,
-0.009496788494288921,
-0.011472582817077637,
0.017522549256682396,
-0.0045284610241651535,
0.021748151630163193,
0.10314507782459259,

-0.0043986644595861435,
-0.012330682948231697,
-0.001672033336944878,
-0.04098689928650856,
0.009929444640874863,
0.003398148575797677,
-0.0027275325264781713,
-0.003699204884469509,
-0.008631478063762188,
-0.0014133413787931204,
-0.012071089819073677,
-0.03106466680765152,
-0.04698638990521431,
-0.026334300637245178,
0.020147325471043587,
-0.014205523766577244,
0.01711873710155487,
-0.0030069558415561914,
-0.020190590992569923,
0.0243152417242527,
0.007968072779476643,
0.016945675015449524,
-0.019368546083569527,
-0.004762816242873669,
0.005126968026161194,
0.014594913460314274,
-0.00870358757674694,
0.0061653414741158485,
-0.043236710131168365,
0.021618355065584183,
0.01540253683924675,
-0.030978135764598846,
-0.04015043377876282,
-0.0004328808863647282,
-0.002720321761444211,
0.01381613314151764,
0.0024030408822000027,
0.021647198125720024,
-0.004903429187834263,
0.015907302498817444,
0.006039150059223175,
-0.009403047151863575,
-0.045082706958055496,
0.012590276077389717,
0.0199454203248024,
-0.004070567432790995,
-0.008357462473213673,
-0.042919427156448364

```
        ],
    },
},
{
    "_index": "github_issues",
    "_id": "A2LQUZMBKaR7I-arfsTD",
    "_score": 1.0,
    "_source": {
        "_type": "issue",
        "_repo": "langchain",
        "_issueNumber": "28275",
        "_title": "docs: poetry publish",
        "_createdAt": "2024-11-21T23:57:01Z",
        "_closedAt": "2024-12-31T00:36:30Z",
        "_state": "open",
        "_body": "None",
        "GitHub_Issue_vector": [
            -0.008065586909651756,
            -0.007394031155854464,
            0.002886995440348983,
            0.0024300608783960342,
            -0.03137617185711861,
            0.017612747848033905,
            -0.014026504941284657,
            -0.028634563088417053,
            -0.019565105438232422,
            -0.009817169047892094,
            0.02715298719704151,
            -0.008093279786407948,
            -0.010073329322040081,
            -0.010509493760764599,
            -0.011527211405336857,
            0.0037627865094691515,
            0.008771758526563644,
            -0.02308211661875248,
            0.01035718247294426,
            0.015438848175108433,
            0.015397308394312859,
            0.02089436911046505,
            -0.020423587411642075,
            0.01256569940596819,
            -0.024051371961832047,
            0.007262489292770624,
            0.017709674313664436,
            -0.022694414481520653,
            0.016505029052495956,
            0.005503984168171883,
            0.015300382860004902,
```

-0.02974228374660015,
0.003006421495229006,
-0.020215891301631927,
0.0006226598052307963,
0.019274327903985977,
0.01866508275270462,
-0.019163556396961212,
0.009727166965603828,
-0.018942013382911682,
-0.001056661014445126,
0.014005735516548157,
0.018415845930576324,
-0.0178204458206892,
-0.01878970116376877,
-0.002855840837582946,
0.016366563737392426,
-0.004873968195170164,
-0.008294053375720978,
0.018332766368985176,
0.0278176199644804,
0.0007459802436642349,
-0.00717941066250205,
0.009429466910660267,
-0.00016345361655112356,
-0.0041747200302779675,
0.01498191338032484,
0.02960381843149662,
-0.003897789865732193,
-0.014102661050856113,
-0.0026914135087281466,
-0.004693963564932346,
-0.017488129436969757,
-0.006874787621200085,
-0.009921018034219742,
-0.011568751186132431,
-0.024355994537472725,
0.02175285294651985,
0.011485672555863857,
-0.0316254086792469,
0.016671186313033104,
0.00875098817050457,
-0.01051641721278429,
-0.028468405827879906,
0.033646997064352036,
-0.013791114091873169,
0.019800495356321335,
-0.014130353927612305,
-0.001589751336723566,

0.005597447976469994,
0.01316802203655243,
-0.009907171130180359,
-0.011783371679484844,
0.029077650979161263,
0.02844071201980114,
0.00600591953843832,
0.030683845281600952,
0.012337232008576393,
0.0040674093179404736,
-0.019094323739409447,
0.02902226522564888,
0.011970299296081066,
0.029049957171082497,
0.0168373454362154,
0.0199112668633461,
0.0396563783288002,
-0.011125663295388222,
0.025657566264271736,
0.03173618018627167,
-0.03636091202497482,
-0.005981688387691975,
0.01629733107984066,
-0.02254210226237774,
-0.010876426473259926,
-0.01937125436961651,
-0.013943426311016083,
0.006244771648198366,
-0.007539419457316399,
0.029299195855855942,
0.010717191733419895,
-0.0045901150442659855,
0.03456086665391922,
0.011312590911984444,
-0.053198255598545074,
0.009782552719116211,
0.0024906392209231853,
-0.010883348993957043,
-0.015965014696121216,
0.0069474815391004086,
-0.0011622406309470534,
0.03594551607966423,
0.012115688063204288,
-0.0009779090760275722,
-0.01219184324145317,
0.010080252774059772,
0.002090821508318186,
-0.003946252632886171,

-0.02731914632022381,
-0.01856815628707409,
-0.011243358254432678,
0.019454333931207657,
0.016768112778663635,
0.010544110089540482,
-0.006064767017960548,
0.0009770436445251107,
0.015688085928559303,
-0.01874816231429577,
0.011603367514908314,
-0.022389791905879974,
-0.014871141873300076,
-0.013410335406661034,
0.052755165845155716,
-0.0258098766207695,
-0.015286536887288094,
-0.02107437327504158,
0.015397308394312859,
0.000698382849805057,
-0.004946662113070488,
0.021683620288968086,
-0.006507855374366045,
0.01950971968472004,
-0.017792753875255585,
0.0009181960485875607,
-0.006760553922504187,
-0.004607423208653927,
0.017155814915895462,
-0.002353905001655221,
0.01955125853419304,
-0.015840396285057068,
0.001416670042090118,
0.015148071572184563,
0.021849777549505234,
0.015217304229736328,
-0.018540464341640472,
0.0010211793705821037,
0.03414547070860863,
0.02741607092320919,
-0.0032660432625561953,
-0.014261895790696144,
-0.004780504386872053,
-0.030296143144369125,
-0.0014876334462314844,
-0.02813608944416046,
0.03048999421298504,
-0.022334406152367592,

0.03805018216371536,
0.0056389872916042805,
0.01579885743558407,
-0.011361053213477135,
-0.004808197263628244,
-0.03852096572518349,
0.01259339228272438,
0.005964380223304033,
0.048241209238767624,
-0.01968972384929657,
0.013624956831336021,
0.01053718663752079,
0.0030427684541791677,
0.022126708179712296,
-0.002800454618409276,
0.010758730582892895,
0.027969930320978165,
-0.001933317631483078,
0.005947072058916092,
-0.6265264749526978,
-0.02432830072939396,
-0.003501433879137039,
0.004693963564932346,
0.006466315593570471,
0.013417258858680725,
0.006002457812428474,
-0.0008277611341327429,
-0.0039670225232839584,
-0.0021254378370940685,
-0.016269637271761894,
0.004918969236314297,
0.015411155298352242,
-0.009263308718800545,
0.002618719357997179,
-0.004406648688018322,
0.025560639798641205,
-0.005407058633863926,
-0.022043628618121147,
0.00270179845392704,
-0.023815980181097984,
0.02129591815173626,
0.017169659957289696,
0.007477110251784325,
-0.002135822782292962,
-0.020437434315681458,
0.025505254045128822,
0.011665676720440388,
-0.005666680168360472,

-0.01575731672346592,
-0.03289928659796715,
0.0169204231351614,
0.0113887470215559,
0.011326437816023827,
0.0494873933494091,
-0.002052743686363101,
-0.0057220663875341415,
0.018720468506217003,
-0.006109768524765968,
0.022251326590776443,
-0.025879109278321266,
-0.0006672282470390201,
0.014234202913939953,
0.000593668723013252,
-0.01404035184532404,
-0.01701734960079193,
0.024480612948536873,
-0.013285716995596886,
0.007657114882022142,
0.015618852339684963,
0.01223383022248745,
0.004437803290784359,
-0.014428053982555866,
-0.00032625815947540104,
0.017155814915895462,
-0.01243415754288435,
0.02844071201980114,
-0.027651460841298103,
-0.0003931107930839062,
0.015120378695428371,
0.006338235456496477,
-0.012087995186448097,
-0.006445546168833971,
-0.012205690145492554,
-0.010883348993957043,
0.011077200062572956,
0.007234796416014433,
0.023151349276304245,
0.0033387374132871628,
-0.03691476956009865,
0.0024612154811620712,
-0.0017143698642030358,
-0.0047355033457279205,
-0.035336270928382874,
0.013071096502244473,
0.009907171130180359,
0.01937125436961651,

0.03126539662480354,
-0.009124843403697014,
0.006826324854046106,
-0.012094918638467789,
-0.01852661743760109,
-0.01509268581867218,
-0.010911042802035809,
0.03361930325627327,
0.011007968336343765,
-0.01589578203856945,
0.0047043487429618835,
-0.000920792284887284,
-0.006269003264605999,
-0.004025870002806187,
0.02173900604248047,
0.002681028563529253,
-0.008716371841728687,
-0.013472644612193108,
0.027609921991825104,
0.006535548251122236,
0.01225415337830782,
0.024189837276935577,
-0.02562987245619297,
-0.0012885899050161242,
0.01571577787399292,
0.004136641975492239,
0.0026533356867730618,
0.026197578758001328,
0.007373261731117964,
-0.03796710446476936,
-0.005053972825407982,
0.01960664428770542,
-0.00480127427726984,
-0.011513365432620049,
0.006172077730298042,
0.0003539511817507446,
0.018498925492167473,
0.007629422005265951,
-0.04447495937347412,
0.01257262285798788,
-0.011547981761395931,
0.005898609291762114,
0.007435570936650038,
-0.007643268443644047,
0.019537411630153656,
-0.012233383022248745,
0.028606869280338287,
0.010419491678476334,

0.01964818499982357,
-0.023829827085137367,
-0.03821634128689766,
-0.02067282609641552,
0.008314823731780052,
-0.009574854746460915,
0.021282071247696877,
0.04140103608369827,
0.008397902362048626,
0.021365150809288025,
0.010149484500288963,
0.022528257220983505,
0.008744064718484879,
0.019412793219089508,
0.01768198050558567,
-0.007615575101226568,
-0.005998996552079916,
0.008709448389708996,
-0.014144199900329113,
0.013264947570860386,
-0.05870916321873665,
-0.02607296034693718,
-0.002057936042547226,
-0.016975808888673782,
0.023455971851944923,
-0.017778906971216202,
-0.006490547209978104,
-0.014732676558196545,
-0.0056389872916042805,
0.0020994755905121565,
0.0017913910560309887,
0.012669548392295837,
-0.0269175972789526,
-0.018554311245679855,
-0.0015101339668035507,
0.012177997268736362,
0.001779275364242494,
-0.014815755188465118,
-0.01422035600990057,
-0.02558833360671997,
-0.016768112778663635,
-0.014968067407608032,
0.0033162368927150965,
0.000845501956064254,
-0.02639142982661724,
0.009436390362679958,
-0.017294278368353844,
0.010281026363372803,

-0.012531083077192307,
-0.0031933491118252277,
-0.007110178004950285,
-0.02022973634302616,
0.007560189347714186,
-0.01840199902653694,
0.00453126709908247,
-0.011838758364319801,
0.006580549292266369,
-0.012946478091180325,
0.0013258024118840694,
0.019080476835370064,
0.007172487210482359,
0.029077650979161263,
0.03505934029817581,
0.0007624229183420539,
-0.007899428717792034,
-0.0134587986394763,
0.016865037381649017,
-0.0034979721531271935,
0.022445177659392357,
-0.02093590795993805,
-0.0015906167682260275,
-0.0144003601744771,
0.001293782377615571,
0.007262489292770624,
0.04638577625155449,
-0.004181643016636372,
0.014261895790696144,
0.01960664428770542,
-0.012870321981608868,
-0.004036254715174437,
-0.023165196180343628,
0.0022483256179839373,
-0.025422174483537674,
0.018069682642817497,
-0.005230515729635954,
0.021683620288968086,
-0.008342516608536243,
-0.015965014696121216,
-0.010945658199489117,
0.00210986053571105,
-0.013140329159796238,
-0.007947891019284725,
-0.03436701372265816,
-0.02227901853621006,
0.0020319740287959576,
-0.017086582258343697,

0.005777452141046524,
0.001306763500906527,
-0.018858933821320534,
0.013341103680431843,
-0.0031362322624772787,
0.014594211243093014,
0.016408102586865425,
0.010467953979969025,
-0.013666495680809021,
-0.006528624799102545,
0.02518678456544876,
0.013888039626181126,
-0.00255121779628098,
0.0001460373168811202,
0.012108764611184597,
0.03987792134284973,
-0.02254210226237774,
0.015840396285057068,
-0.016366563737392426,
-0.0019194710766896605,
0.012967247515916824,
-0.0026325657963752747,
-0.02669605240225792,
0.014857294969260693,
-0.0018831241177394986,
0.019191250205039978,
-0.007920198142528534,
0.0016382141038775444,
-4.1025665268534794e-05,
-0.0078025031834840775,
-0.007940968498587608,
-0.006930173374712467,
-0.022306712344288826,
0.016449643298983574,
-0.0228190328925848,
-0.018332766368985176,
0.019495872780680656,
-0.01643579639494419,
0.03528088331222534,
-0.0026948752347379923,
-0.010419491678476334,
0.012524159625172615,
0.037662480026483536,
0.03112693317234516,
-0.017335819080471992,
-0.009741012938320637,
-0.018055835738778114,
-0.003565473947674036,

0.002255248837172985,
-0.009713320061564445,
-0.011367976665496826,
0.007809426169842482,
-0.022126708179712296,
0.036997850984334946,
0.011367976665496826,
0.015051146037876606,
0.010592573322355747,
0.02835763245820999,
-0.0024231376592069864,
-0.048905838280916214,
-0.004330493044108152,
0.005483214277740955,
0.015064992941915989,
0.011492595076560974,
0.0005664083873853087,
-0.018208147957921028,
0.01187337376177311,
-0.027291452512145042,
0.009609471075236797,
0.025505254045128822,
-0.010170254856348038,
-0.002501024166122079,
-0.010620266199111938,
-0.01638041064143181,
-0.022472869604825974,
0.02625296451151371,
-0.008917146362364292,
0.00207697506994009,
0.020645132288336754,
-0.003534319344907999,
0.009104073978960514,
0.03159771487116814,
0.009706396609544754,
0.028689948841929436,
0.006497470196336508,
0.017446590587496758,
-0.02326212078332901,
-0.005818991921842098,
-0.015286536887288094,
0.03129309043288231,
0.01053718663752079,
0.006265541538596153,
-0.020077425986528397,
-0.0189697053283453,
0.007664037868380547,
-0.0011397401103749871,

-0.010467953979969025,
0.011776448227465153,
-0.012911861762404442,
-0.012752627022564411,
-0.034090083092451096,
-0.0007178545347414911,
-0.005258208606392145,
0.057933758944272995,
0.045610371977090836,
0.019315868616104126,
0.007906352169811726,
-0.006421314552426338,
0.008896376937627792,
-0.029936134815216064,
-0.011748755350708961,
0.005981688387691975,
-0.011035661213099957,
-0.012814936228096485,
-0.008660986088216305,
0.004960509017109871,
-0.002914688317105174,
0.02410675771534443,
-0.0038320189341902733,
-0.006314004305750132,
-0.035779356956481934,
-0.011146432720124722,
-0.0010298334527760744,
0.009526392444968224,
-0.033813152462244034,
-0.0004184239369351417,
0.0016581184463575482,
0.02916073054075241,
-0.00607169046998024,
0.0010523340897634625,
0.007774809841066599,
0.013396489433944225,
-0.01125028170645237,
-0.0016806189669296145,
-0.006487085483968258,
0.0062932344153523445,
0.018097376450896263,
-0.015051146037876606,
0.05854300409555435,
0.014441899955272675,
0.004029331728816032,
0.013223407790064812,
0.018554311245679855,
0.022832879796624184,

0.010834886692464352,
-0.0009406966273672879,
-0.031154626980423927,
-0.005767067428678274,
-0.013811884447932243,
-0.023275967687368393,
0.03924098238348961,
-0.0020423587411642075,
-0.01955125853419304,
0.03461625054478645,
0.013915733434259892,
-0.03749632462859154,
-0.029548432677984238,
0.026806825771927834,
0.029271502047777176,
-0.015328075736761093,
0.020423587411642075,
-0.022306712344288826,
-0.020063579082489014,
-0.014857294969260693,
0.004873968195170164,
-0.016131173819303513,
-0.011790295131504536,
-0.007643268443644047,
0.0014564787270501256,
-0.016463488340377808,
-0.008356363512575626,
-0.02571295201778412,
0.016352716833353043,
0.015051146037876606,
-0.001385515439324081,
0.006279387976974249,
-0.012994940392673016,
0.017557362094521523,
0.0016399448504671454,
0.02629450522363186,
-0.01125028170645237,
0.0019108171109110117,
0.008654062636196613,
-0.030877696350216866,
-0.032123882323503494,
0.021725159138441086,
-0.014303434640169144,
0.013431105762720108,
-0.0032885437831282616,
-0.009657934308052063,
-0.007065176963806152,
-0.015729624778032303,

0.03220696002244949,
-0.019675876945257187,
0.0021877470426261425,
-0.005895147565752268,
-0.031043853610754013,
0.02472984977066517,
-0.003262581769376993,
0.0067709386348724365,
0.029437659308314323,
-0.0139295794069767,
0.020202044397592545,
0.0031950799748301506,
-0.03996099904179573,
-0.03276082128286362,
-0.03018537163734436,
-0.010412568226456642,
-0.00787865836173296,
-0.0033231601119041443,
0.02508985809981823,
-0.023815980181097984,
0.024051371961832047,
0.015508080832660198,
-0.009235615842044353,
0.010710268281400204,
0.0001552322501083836,
0.0018692775629460812,
0.01319571491330862,
0.01011486817151308,
0.0006360736442729831,
-0.010384875349700451,
-0.0031777718104422092,
0.0019315867684781551,
-0.022292865440249443,
0.024120604619383812,
0.03755170851945877,
-0.00028818027931265533,
0.032262347638607025,
-0.007470186799764633,
-0.024937547743320465,
-0.012780319899320602,
0.008245591074228287,
0.017709674313664436,
0.018318919464945793,
0.022735953330993652,
-0.015078838914632797,
-0.022791339084506035,
-0.028634563088417053,
0.016145018860697746,

0.003949714358896017,
-0.018235841765999794,
-0.007580959238111973,
-0.011007968336343765,
-0.007477110251784325,
-0.02893918566405773,
-0.02616988681256771,
0.02219594083726406,
-0.03508703410625458,
-0.005683988332748413,
-0.0017161006107926369,
0.01225415337830782,
0.006566702853888273,
-0.01643579639494419,
-0.00021883961744606495,
0.008183281868696213,
0.0017308125970885158,
-0.00980332214385271,
0.001262627774849534,
-0.0030168062075972557,
-0.016463488340377808,
0.03738555312156677,
0.015161918476223946,
0.04702271521091461,
0.002738145412877202,
0.029437659308314323,
-0.001328398589976132,
0.026460662484169006,
-0.007117101456969976,
0.004849736578762531,
-0.004347801208496094,
-0.010606419295072556,
-0.0038147110026329756,
0.015106531791388988,
0.017072735354304314,
0.011395669542253017,
-0.0023954445496201515,
-0.006466315593570471,
0.010412568226456642,
-0.017155814915895462,
-0.00035568198654800653,
-0.03320390731096268,
-0.0377732515335083,
-0.02334520034492016,
-0.014428053982555866,
-0.020866677165031433,
0.017128121107816696,
-0.03209618851542473,

-0.01665733940899372,
0.04333954676985741,
-0.0015923475148156285,
-0.0015386923914775252,
-0.009561008773744106,
0.025200631469488144,
-0.01884508691728115,
0.012835705652832985,
0.005171667784452438,
0.01317494548857212,
-0.023691361770033836,
0.01589578203856945,
-0.019855881109833717,
-0.02330365963280201,
0.0024819851387292147,
0.010197947733104229,
0.0015637892065569758,
0.0004963105311617255,
0.011942606419324875,
-0.03245619684457779,
0.04519497603178024,
-0.009927940554916859,
-0.011077200062572956,
-0.02147592231631279,
-0.01808352954685688,
-0.006950943265110254,
-0.01524499710649252,
0.00505743408575654,
-0.033951617777347565,
0.022002089768648148,
0.01728043332695961,
0.007096331566572189,
0.01808352954685688,
-0.016394255682826042,
-0.011631060391664505,
0.0057012964971363544,
-0.01334802620112896,
0.03458855673670769,
0.02946535311639309,
0.015189611352980137,
0.023871367797255516,
0.00672939931973815,
-0.02817762829363346,
0.015521926805377007,
-0.022085167467594147,
-0.00474588805809617,
0.01884508691728115,
0.009921018034219742,

0.009387927129864693,
-0.020963601768016815,
0.02500678040087223,
-2.3082007828634232e-05,
-0.013846500776708126,
-0.03780094534158707,
0.02960381843149662,
-0.011056430637836456,
0.01256569940596819,
-0.02232055924832821,
-0.00195927987806499,
-0.017751213163137436,
0.020340509712696075,
-0.03508703410625458,
0.006725937593728304,
-0.023372892290353775,
0.0009874285897240043,
-0.0002475061919540167,
0.010841810144484043,
-0.002708721673116088,
0.036499373614788055,
0.0008645408670417964,
-0.02245902456343174,
-0.00475627277046442,
0.005756682716310024,
-0.02446676604449749,
0.01759890280663967,
0.006535548251122236,
0.04569345340132713,
-0.014275741763412952,
-0.006442084442824125,
0.028468405827879906,
-0.0015404231380671263,
0.004700887016952038,
-0.008889453485608101,
-0.0039012515917420387,
0.033065441995859146,
-0.02772069349884987,
0.00816943496465683,
-0.003061807481572032,
-0.01064795907586813,
0.0036416295915842056,
-0.02380213513970375,
-0.009297925047576427,
-0.030656151473522186,
-0.017128121107816696,
-0.00391855975612998,
0.005126666743308306,

0.019038937985897064,
-0.014151123352348804,
-0.012808012776076794,
-0.020243583247065544,
-0.01964818499982357,
-0.00935331080108881,
-0.009907171130180359,
0.012738780118525028,
-0.03134847804903984,
-0.032483890652656555,
-0.004036254715174437,
-0.043422624468803406,
0.013105712831020355,
-0.002947573782876134,
0.016588106751441956,
-0.015646545216441154,
0.04951508715748787,
-0.020160505548119545,
0.02647450938820839,
-0.01946817897260189,
-0.04638577625155449,
-0.011181049048900604,
0.004742426332086325,
-0.02366366982460022,
-0.01656041480600834,
-0.004018946550786495,
-0.010675651952624321,
-0.022043628618121147,
0.010675651952624321,
0.0023071730975061655,
0.0035222035367041826,
-0.020118964836001396,
-0.013278793543577194,
0.017668135464191437,
0.01509268581867218,
0.006680936552584171,
-0.009381003677845001,
-0.001220222911797464,
0.04015485197305679,
-0.01856815628707409,
-0.0006278522778302431,
-0.0054001351818442345,
-0.006573626305907965,
-0.01466344390064478,
-0.0019229328026995063,
0.027028368785977364,
-0.017114274203777313,
-0.006123614963144064,

-0.02456369251012802,
0.007574035786092281,
-0.001177817932330072,
-0.010834886692464352,
-0.02330365963280201,
-0.010758730582892895,
-0.03032383695244789,
-0.03622244670987129,
0.017349665984511375,
0.02514524571597576,
0.007110178004950285,
0.014580365270376205,
0.015840396285057068,
0.02669605240225792,
0.015314229764044285,
-0.02133745700120926,
-0.020243583247065544,
-0.03575166314840317,
-0.008702525869011879,
-0.001457344158552587,
-0.02857917733490467,
-0.01995280757546425,
0.03450547903776169,
0.03417316451668739,
0.009727166965603828,
-0.017405051738023758,
-0.011866451241075993,
-0.0024023677688091993,
-0.027845311909914017,
-0.015868090093135834,
-0.014760369434952736,
0.001056661014445126,
0.0008104529697448015,
0.004043178167194128,
-0.0003954906715080142,
-0.004635116085410118,
0.01066872850060463,
0.0008853106410242617,
-0.021143605932593346,
-0.0009432928636670113,
0.0013050326379016042,
0.014303434640169144,
-0.02968689799308777,
-0.009104073978960514,
-0.01096642855548191,
0.02468831092119217,
0.025920648127794266,
0.010502570308744907,

0.010938735678792,
-0.01109797041863203,
-0.02446676604449749,
-0.03289928659796715,
-0.023137502372264862,
-0.0013664765283465385,
0.01080719381570816,
0.02027127705514431,
-0.0336746908724308,
-0.021877471357584,
0.0016892730491235852,
-0.010613342747092247,
0.009083304554224014,
-0.02767915464937687,
0.01996665447950363,
0.017668135464191437,
0.038825586438179016,
-0.03032383695244789,
-0.008044816553592682,
-0.004700887016952038,
-0.01611732691526413,
0.027305299416184425,
0.0024906392209231853,
-0.003236619522795081,
0.02603142149746418,
-0.015411155298352242,
-0.003032383508980274,
-0.012524159625172615,
0.03129309043288231,
-0.022735953330993652,
-0.00511282030493021,
0.02737453207373619,
0.023386739194393158,
0.006525163538753986,
-0.015785010531544685,
-0.0037939411122351885,
-0.0257683377712965,
0.0031656562350690365,
0.0074563403613865376,
0.01553577370941639,
-0.007345568388700485,
-0.004361647646874189,
0.007366338279098272,
-0.042176440358161926,
-0.023857520893216133,
0.002674105344340205,
0.020520513877272606,
0.0024248682893812656,

-0.00922176893800497,
-0.015452694147825241,
-0.002608334645628929,
-0.00904176477342844,
0.00888253003358841,
-0.008924069814383984,
-0.02902226522564888,
-0.008141742087900639,
0.011104893870651722,
-0.013258024118840694,
0.023109808564186096,
0.2030450850725174,
-0.006725937593728304,
0.020257430151104927,
0.03004690632224083,
-0.006348620634526014,
0.04045947268605232,
-0.0046835788525640965,
-0.0070720999501645565,
0.0039670225232839584,
0.01710042729973793,
0.018235841765999794,
-0.01422035600990057,
0.0011406054254621267,
-0.011513365432620049,
-0.007691731210798025,
0.025394482538104057,
-0.03173618018627167,
-0.02982536144554615,
-0.024799082428216934,
-0.02348366566002369,
-0.01316802203655243,
-0.0109595051035285,
-0.0024854468647390604,
-0.004171258304268122,
0.008688678964972496,
-0.000293589080683887,
-0.010315642692148685,
-0.00886176060885191,
-0.010461031459271908,
0.015120378695428371,
-0.008543291129171848,
0.014635751023888588,
0.010585649870336056,
-0.018775854259729385,
-0.0033145060297101736,
0.0030843080021440983,
0.030656151473522186,

0.014732676558196545,
0.030573073774576187,
0.002082167426124215,
0.0084394421428442,
-0.02731914632022381,
0.024009831249713898,
-0.0049224309623241425,
0.014732676558196545,
0.04281337931752205,
-0.013804960995912552,
-0.017778906971216202,
-0.009699473157525063,
0.008217898197472095,
-0.022126708179712296,
-0.008660986088216305,
0.051093585789203644,
0.02910534478724003,
0.02111591398715973,
-0.001902163028717041,
-0.011984146200120449,
0.025103705003857613,
0.0020077426452189684,
0.012731857597827911,
-0.01937125436961651,
0.0143588213250041,
0.005220130551606417,
0.016588106751441956,
-0.02526986412703991,
-0.01898355223238468,
-0.039407141506671906,
-0.025242170318961143,
0.003427008865401149,
-0.018291227519512177,
-0.009069457650184631,
-0.022749800235033035,
0.010301796719431877,
0.0055143688805401325,
-0.005438213236629963,
-0.02549140714108944,
0.011554904282093048,
0.0258098766207695,
0.016311177983880043,
0.033646997064352036,
0.0020735133439302444,
-0.004448188468813896,
-0.013244178146123886,
-0.016061941161751747,
0.0031396939884871244,

-0.039545606821775436,
0.0011293551651760936,
0.009581778198480606,
-0.022154400125145912,
-0.010343335568904877,
-0.020506666973233223,
-0.010197947733104229,
-0.0015248458366841078,
-0.004240490961819887,
0.016892731189727783,
0.011367976665496826,
0.0018017758848145604,
0.010994121432304382,
-0.00612707668915391,
0.0024542922619730234,
-0.027665307745337486,
0.04322877526283264,
0.008937915787100792,
-0.006660166662186384,
-0.015064992941915989,
0.007504803128540516,
0.0054312897846102715,
-0.016145018860697746,
-0.01870662160217762,
-0.026765285059809685,
0.026183731853961945,
-0.022168247029185295,
-0.002343520289286971,
0.014746523462235928,
-0.004333954770117998,
0.00904176477342844,
0.010156407952308655,
-0.025519100949168205,
0.014248048886656761,
0.01243415754288435,
0.011174125596880913,
-0.026723746210336685,
0.0021548615768551826,
0.013154175132513046,
0.004479343071579933,
0.0028194936458021402,
-0.012669548392295837,
0.00156032748054713,
-0.002824686001986265,
-0.01406804472208023,
0.0198281891644001,
-0.010731037706136703,
0.0005265997024253011,

-0.03215157240629196,
-0.002234478946775198,
0.025117551907896996,
-0.008273283950984478,
0.003072192193940282,
0.02219594083726406,
-0.014261895790696144,
-0.003023729659616947,
-0.003683169139549136,
-0.014884987846016884,
-0.018817394971847534,
-0.010641035623848438,
-0.016865037381649017,
0.028966879472136497,
-0.014968067407608032,
0.0006391025381162763,
-0.02594834193587303,
-0.020949754863977432,
-0.020686671137809753,
-0.01285647600889206,
-0.006265541538596153,
0.04289645701646805,
-0.031182318925857544,
-0.019398946315050125,
-0.03863173723220825,
-0.023995986208319664,
0.01719735376536846,
-0.03544704243540764,
0.0052132075652480125,
0.0249098539352417,
-0.016643494367599487,
-0.03309313580393791,
-0.0002715212176553905,
-0.1787860095500946,
0.00630708085373044,
0.03658245503902435,
-0.011499518528580666,
0.014130353927612305,
0.012378771789371967,
0.016726572066545486,
-0.018374307081103325,
-0.000490685342811048,
-0.019745109602808952,
0.028745334595441818,
-0.00534821068868041,
-0.007304029073566198,
-0.022583642974495888,
-0.026529895141720772,

0.023608284071087837,
-0.018817394971847534,
0.007044407073408365,
0.01589578203856945,
0.027651460841298103,
0.025242170318961143,
-0.016574261710047722,
0.02317904122173786,
-0.00229332665912807,
0.006757092196494341,
-0.011935683898627758,
-0.011624136939644814,
0.03594551607966423,
-0.0027242989744991064,
-0.010703344829380512,
-0.011395669542253017,
0.012517236173152924,
0.011811064556241035,
-0.012081071734428406,
0.0376070961356163,
0.016629647463560104,
0.0030843080021440983,
-0.00920099951326847,
0.015175764448940754,
0.028773028403520584,
-0.006570164579898119,
0.031459249556064606,
0.011644907295703888,
-0.01996665447950363,
-0.019454333931207657,
0.01696196384727955,
0.016186559572815895,
0.010343335568904877,
0.01009409874677658,
-0.03818864747881889,
-0.002582372399047017,
-0.019260482862591743,
-0.016491182148456573,
0.0025789106730371714,
0.006937096826732159,
0.023054422810673714,
0.006590934470295906,
0.011354130692780018,
0.002355635864660144,
-0.019745109602808952,
-0.0043824175372719765,
-0.025699105113744736,
0.0075324964709579945,

-0.0044966512359678745,
-0.0021202454809099436,
0.004043178167194128,
-0.023497510701417923,
0.03317621350288391,
-0.0006222270894795656,
-0.0014971528435125947,
-0.018831240013241768,
-0.0039358679205179214,
0.01914970949292183,
0.013569570146501064,
0.011776448227465153,
-0.0034062392078340054,
-0.00667401310056448,
0.017446590587496758,
0.01669888012111187,
0.0143588213250041,
-0.014344974420964718,
0.025602178648114204,
-0.02330365963280201,
0.0023089039605110884,
0.03960099071264267,
0.008093279786407948,
-0.005787837319076061,
-0.012288768775761127,
-0.0002574583631940186,
0.014732676558196545,
0.021420536562800407,
-0.018360460177063942,
0.003873558482155204,
0.0005741970962844789,
0.011084123514592648,
0.011755678802728653,
0.001488498761318624,
0.0030479608103632927,
0.0114095164462924,
-0.0009303117403760552,
0.009616394527256489,
0.012524159625172615,
-0.021448228508234024,
0.0024023677688091993,
0.03486548736691475,
-0.012981094419956207,
-0.01579885743558407,
0.026239117607474327,
0.037440937012434006,
-0.018069682642817497,
-0.022666720673441887,

0.026363737881183624,
0.008224821649491787,
0.018997399136424065,
0.009471006691455841,
0.019994346424937248,
0.023677516728639603,
-0.02201593667268753,
0.011451056227087975,
-0.0018554311245679855,
0.03411777690052986,
-0.027914544567465782,
-0.016172712668776512,
0.012821859680116177,
-0.008072509430348873,
-0.002890456933528185,
-0.054444439709186554,
-0.018249686807394028,
-0.00044741504825651646,
-0.0031085393857210875,
-0.03048999421298504,
-0.0029925748240202665,
-0.00814866553992033,
0.03893635794520378,
0.007158640772104263,
-0.010772577486932278,
-0.00022803455067332834,
0.017529670149087906,
0.00629669614136219,
0.013479568064212799,
0.021669773384928703,
-0.020326662808656693,
0.002758915303274989,
-0.008501751348376274,
-0.01852661743760109,
0.011748755350708961,
0.010862579569220543,
-0.02380213513970375,
0.008515598252415657,
0.0017230239463970065,
0.0024283300153911114,
0.027166834101080894,
-0.03303775191307068,
0.02639142982661724,
0.016048094257712364,
-0.010917965322732925,
0.021365150809288025,
-0.019800495356321335,
-0.011804142035543919,

-0.015064992941915989,
-0.0055766780860722065,
0.018291227519512177,
-0.003122385824099183,
-0.012697241269052029,
0.017155814915895462,
-0.03652706742286682,
-0.009879478253424168,
0.044308800250291824,
0.02916073054075241,
-0.037440937012434006,
-0.008543291129171848,
-0.014524978585541248,
0.0023712131660431623,
0.015175764448940754,
0.0019004321657121181,
-0.02263902872800827,
-0.048158127814531326,
0.01812506839632988,
-0.020645132288336754,
-0.008162512443959713,
0.005472829099744558,
-0.016892731189727783,
-0.0024577537551522255,
0.03757940232753754,
0.0017714867135509849,
-0.02799762412905693,
0.0010627189185470343,
-0.023774441331624985,
-0.0009484852780587971,
0.030157677829265594,
0.01020487118512392,
0.004752811510115862,
0.0013015710283070803,
-0.029797669500112534,
0.010371028445661068,
-0.013576493598520756,
-0.015106531791388988,
0.014026504941284657,
-0.01111873984336853,
0.007719424087554216,
-0.019191250205039978,
-0.024037525057792664,
-0.0007481437060050666,
-0.009962556883692741,
0.010308719240128994,
-0.010038712993264198,
-0.01538346242159605,

-0.010142561979591846,
0.00020391136058606207,
-0.0009969479870051146,
0.010412568226456642,
0.015965014696121216,
0.016352716833353043,
0.024217529222369194,
0.02938227355480194,
-0.020742056891322136,
0.012247229926288128,
0.015078838914632797,
0.032483890652656555,
-0.04652424156665802,
0.005798222031444311,
0.007947891019284725,
0.01719735376536846,
-0.002473331056535244,
-0.005299747921526432,
0.0026066037826240063,
-0.04242567718029022,
-0.03137617185711861,
-0.08817451447248459,
0.026682207360863686,
-0.0033127751667052507,
-0.04477958381175995,
-0.014248048886656761,
-0.03126539662480354,
0.012025685980916023,
-0.01674041897058487,
0.00022565468680113554,
0.005881301127374172,
-0.009360234253108501,
0.01910817064344883,
0.011201818473637104,
-0.01538346242159605,
-0.012226459570229053,
-0.017875831574201584,
0.018138915300369263,
-0.011956453323364258,
-0.004804735537618399,
0.01069642137736082,
-0.009858707897365093,
-0.020368201658129692,
0.002971805166453123,
0.011637983843684196,
-0.013811884447932243,
0.014234202913939953,
-0.0160204004496336,

0.03411777690052986,
-0.0014971528435125947,
0.0011033930350095034,
0.009983327239751816,
-0.04616423323750496,
-0.01469113677740097,
0.018540464341640472,
0.019232789054512978,
0.013084943406283855,
-0.002672374714165926,
0.025103705003857613,
0.013742651790380478,
0.02857917733490467,
-0.01741889677941799,
-0.018138915300369263,
0.02236209809780121,
-0.03544704243540764,
0.0010081983637064695,
0.02593449503183365,
-0.008051740005612373,
-0.00157677021343261,
0.018471231684088707,
-0.004946662113070488,
0.036499373614788055,
0.027665307745337486,
-0.013943426311016083,
-0.02456369251012802,
0.005088589154183865,
-0.009824092499911785,
0.004607423208653927,
0.02643296867609024,
-0.012773396447300911,
-0.013424182310700417,
0.02468831092119217,
0.001663310918956995,
0.006088998634368181,
-0.018235841765999794,
0.019634338095784187,
-0.007643268443644047,
-0.0034945106599479914,
0.0011657022405415773,
0.0034650866873562336,
-0.0396563783288002,
-0.021988242864608765,
-0.02175285294651985,
-0.004995124880224466,
0.003184695029631257,
0.023899059742689133,

-0.010197947733104229,
-0.0003031085361726582,
-0.009962556883692741,
0.007345568388700485,
0.034007005393505096,
0.005147436633706093,
-0.004372032359242439,
-0.005282439757138491,
0.014649597927927971,
0.027845311909914017,
0.013029556721448898,
-0.0072486428543925285,
0.004115872085094452,
0.004569345153868198,
-0.005936686880886555,
-0.03857634961605072,
0.004697425290942192,
-0.0031466172076761723,
0.020908216014504433,
0.0034425861667841673,
0.02508985809981823,
-0.02693144418299198,
0.010232564061880112,
-0.006757092196494341,
-0.006466315593570471,
0.012988017871975899,
-0.00044611693010665476,
0.017308125272393227,
-0.023012883961200714,
-0.02751299738883972,
0.004867044743150473,
-0.03406239300966263,
-0.010017943568527699,
-0.004399725701659918,
0.01701734960079193,
-0.00018108625954482704,
-0.004552036989480257,
-0.015701930969953537,
-0.004489727783948183,
-0.01790352538228035,
0.01129874400794506,
0.003333545057103038,
-0.006404006388038397,
-0.024660617113113403,
0.035557813942432404,
0.006421314552426338,
0.01581270433962345,
0.047659654170274734,

-0.028108395636081696,
0.009367157705128193,
0.02358059026300907,
-0.013728804886341095,
0.003423547139391303,
0.029991520568728447,
-0.005680526606738567,
0.01170029304921627,
-0.007816349156200886,
-0.024175990372896194,
-0.008384056389331818,
-0.0015923475148156285,
-0.007207103539258242,
-0.030960775911808014,
-0.0055455234833061695,
0.0002719539334066212,
0.08662370592355728,
0.008688678964972496,
-0.010419491678476334,
0.011901067569851875,
-0.029271502047777176,
0.018055835738778114,
-0.007816349156200886,
-0.0012011838844045997,
0.007767886854708195,
-0.025339096784591675,
-0.0028039163444191217,
-0.007352491840720177,
-0.02155900187790394,
-0.050068944692611694,
-0.024480612948536873,
0.006760553922504187,
-0.017045041546225548,
0.01037795189768076,
-0.0027416071388870478,
-0.0016347524942830205,
0.029769975692033768,
0.021351303905248642,
0.002379867248237133,
-0.004372032359242439,
-0.026737593114376068,
0.00687824934720993,
0.022182093933224678,
-0.00980332214385271,
0.0016174443298950791,
-0.010661805048584938,
-0.0007291047950275242,
-0.0037974028382450342,

```

        -0.010232564061880112,
        -0.014649597927927971,
        0.009713320061564445,
        0.02089436911046505,
        0.005801683757454157,
        -0.0036243214271962643,
        0.008709448389708996,
        0.0026325657963752747,
        0.009782552719116211,
        0.015452694147825241,
        -0.04372724890708923,
        -0.029216116294264793,
        0.01020487118512392,
        -0.026640666648745537,
        0.005881301127374172,
        -0.012157227843999863,
        -0.029133036732673645
    ],
}
},
{
    "_index": "github_issues",
    "_id": "BGLQUZMBKaR7I-arfsTD",
    "_score": 1.0,
    "_source": {
        "_type": "issue",
        "_repo": "langchain",
        "_issueNumber": "28274",
        "_title": "docs: Update langchain docs to new Databricks package",
        "_createdAt": "2024-11-21T23:51:30Z",
        "_closedAt": "2024-12-31T00:36:30Z",
        "_state": "open",
        "_body": "Thank you for contributing to LangChain!\\r\\n\\r\\nCtrl+F to find instances of `langchain-databricks` and replace with `databricks-langchain`.\\r\\n\\r\\nAdditional guidelines:\\r\\n- Make sure optional dependencies are imported within a function.\\r\\n- Please do not add dependencies to pyproject.toml files (even optional ones) unless they are required for unit tests.\\r\\n- Most PRs should not touch more than one package.\\r\\n- Changes should be backwards compatible.\\r\\n- If you are adding something to community, do not re-import it in langchain.\\r\\n\\r\\nIf no one reviews your PR within a few days, please @-mention one of baskaryan, efriis, eyurtsev, ccurme, vbarada, hwchase17.\\r\\n",
        "GitHub_Issue_vector": [
            -0.011543878354132175,
            0.004838990513235331,
            -0.009220581501722336,
            -0.03720178082585335,
            -0.007434547878801823,
            0.03360066935420036,

```

0.00372453429736197,
-0.017932942137122154,
-0.023276524618268013,
-0.014658547006547451,
0.00908263586461544,
-0.005986117757856846,
-0.011289766989648342,
0.018760615959763527,
0.019326919689774513,
0.00454131793230772,
0.014041421003639698,
-0.01091949176043272,
0.014150326140224934,
-0.0024576117284595966,
-0.02011103183031082,
-0.0037045683711767197,
-0.0022325424943119287,
0.003042066004127264,
-0.01336621306836605,
0.019689934328198433,
0.021461447700858116,
-0.029970521107316017,
-0.00038343461346812546,
0.01553704310208559,
0.008378386497497559,
0.0001802596525521949,
-0.002924086060374975,
-0.010033735074102879,
0.014542382210493088,
-0.014774711802601814,
0.025323929265141487,
-0.004635701887309551,
0.014585943892598152,
0.006697627250105143,
0.02342172898352146,
0.007565233390778303,
0.01292333472520113,
-0.016088826581835747,
-0.020299799740314484,
0.015304713509976864,
0.005158443469554186,
-0.02114199474453926,
-0.010382229462265968,
0.010142640210688114,
0.024075156077742577,
0.027022838592529297,
-0.018847741186618805,
0.004377961158752441,

-0.0070570120587944984,
-0.015043342486023903,
0.02509159967303276,
0.040802888572216034,
-0.0025193244218826294,
-0.01725047454237938,
-0.0014048682060092688,
-0.00487529207020998,
-0.019094590097665787,
0.004784537944942713,
-0.007819343358278275,
-0.009525514207780361,
-0.005398033652454615,
0.03365875408053398,
0.000995568698272109,
0.012596621178090572,
0.023320086300373077,
-0.004915223456919193,
0.013155664317309856,
-0.009707021526992321,
0.02523680403828621,
-0.020517610013484955,
-0.0012787204468622804,
-0.008748662658035755,
0.002163569675758481,
0.008284002542495728,
0.02609352022409439,
0.008937429636716843,
0.023958992213010788,
0.017018144950270653,
0.016001703217625618,
0.015072383917868137,
-0.0011371446307748556,
0.004421522840857506,
-0.03496560826897621,
-0.015725811943411827,
-0.004196453839540482,
-0.018295956775546074,
0.01370018720626831,
-0.011732645332813263,
0.009569075889885426,
0.012756348587572575,
0.007804823108017445,
0.018789658322930336,
0.0026209685020148754,
-0.024351047351956367,
-0.006737559102475643,
0.022332685068249702,

-0.029360655695199966,
-0.01028058584779501,
-0.006320091895759106,
-0.0018305033445358276,
0.024685021489858627,
-0.0038515895139425993,
0.013083062134683132,
0.023058714345097542,
-0.013039499521255493,
0.04893442615866661,
-0.009721542708575726,
-0.037405069917440414,
-0.00012660148786380887,
-0.020401444286108017,
-0.012371552176773548,
-0.03836342692375183,
0.010672641918063164,
0.007841124199330807,
0.03839246928691864,
0.0152611518278718,
0.00047055823961272836,
-0.0010418531019240618,
0.03397820517420769,
0.015391836874186993,
-0.010846888646483421,
-0.007376465480774641,
-0.004704674705862999,
-0.004421522840857506,
0.014426217414438725,
0.018731575459241867,
0.0054887873120605946,
-0.02472858317196369,
-0.026049958541989326,
0.03513985499739647,
-0.012640183791518211,
-0.010004694573581219,
-0.007071532774716616,
0.00170072540640831,
-0.005688445642590523,
0.02706640027463436,
-0.003858849871903658,
-0.021998710930347443,
-0.02806832268834114,
0.013801831752061844,
0.004969675559550524,
0.011050177738070488,
0.018876781687140465,
-0.022013232111930847,

0.02375570312142372,
-0.04176124930381775,
0.018266916275024414,
-0.01998034678399563,
0.005917144939303398,
0.007035231217741966,
0.00562673294916749,
0.003964124247431755,
-0.007347424048930407,
0.0009066300117410719,
-0.00042200495954602957,
0.014585943892598152,
0.01755540631711483,
-0.017235953360795975,
0.007899207063019276,
0.02904120273888111,
0.017119789496064186,
-0.021766381338238716,
-0.009198800660669804,
0.0031745664309710264,
-0.012683745473623276,
0.007645096629858017,
-0.028649145737290382,
0.00943113025277853,
-0.006853723898530006,
0.010694422759115696,
0.026296809315681458,
0.015754852443933487,
-0.013692926615476608,
0.013250048272311687,
-0.012248126789927483,
-0.0009293184848502278,
0.0358368456363678,
0.04698866605758667,
-0.013925256207585335,
0.013344432227313519,
0.01256032008677721,
-0.00855263415724039,
0.014353614300489426,
-0.012952376157045364,
0.02907024323940277,
0.026805030182003975,
-0.015667729079723358,
-0.012625662609934807,
-0.5850059986114502,
-0.023654058575630188,
-0.005721116904169321,
-0.004131110850721598,

-0.01256032008677721,
-0.0005336321191862226,
0.005448855459690094,
0.03046422079205513,
-0.00794276874512434,
0.05157717689871788,
-0.022986112162470818,
0.004595770500600338,
0.009213321842253208,
-0.006737559102475643,
-0.03107408620417118,
-0.009213321842253208,
0.023697620257735252,
-0.02503351680934429,
-0.009293184615671635,
0.011594699695706367,
-0.024888310581445694,
0.034791361540555954,
0.004940634593367577,
-0.018368560820817947,
0.011398672126233578,
0.00856715440750122,
0.012589361518621445,
-0.0019421303877606988,
0.023712141439318657,
-0.0029386067762970924,
-0.028358733281493187,
0.006955367978662252,
0.005779199302196503,
-0.026471056044101715,
0.04681441932916641,
-0.002718982519581914,
-0.013126623816788197,
0.011631001718342304,
-0.00043153410661034286,
0.008683319203555584,
-0.030580386519432068,
-0.012269907630980015,
-0.0014166662003844976,
0.005826391279697418,
0.0007505335961468518,
-0.011413192376494408,
0.023363647982478142,
-0.003312512068077922,
-0.018963905051350594,
-0.008037152700126171,
0.013402515091001987,
0.002606448018923402,

-0.00628742016851902,
-0.007256670389324427,
-0.012828950770199299,
-0.015072383917868137,
0.018049107864499092,
-0.017264995723962784,
0.019094590097665787,
0.011246205307543278,
0.004563098773360252,
0.009358527138829231,
0.005042278673499823,
-0.021548572927713394,
-0.011689083650708199,
-0.018789658322930336,
-0.0025665161665529013,
-0.012988678179681301,
0.016597047448158264,
-0.027661744505167007,
0.005684815347194672,
-0.0041855634190142155,
-0.0006252934108488262,
-0.01646636240184307,
0.00888660829514265,
-0.0030293604359030724,
-0.004356180317699909,
0.018266916275024414,
-0.017177870497107506,
0.04481057450175285,
0.006690367124974728,
-0.019994867965579033,
-0.02681955136358738,
-0.030057644471526146,
0.03847959265112877,
0.016393758356571198,
0.0027117221616208553,
-0.019733497872948647,
-0.02645653486251831,
-0.012763608247041702,
0.0014629506040364504,
0.004334399476647377,
-0.003982274793088436,
-0.0391765832901001,
0.01621951162815094,
0.04033822938799858,
0.018237875774502754,
-0.007013450376689434,
0.014491559937596321,
-0.024583376944065094,

-0.021054871380329132,
0.019791578873991966,
-0.0033306628465652466,
-0.0010581888491287827,
0.03502368927001953,
0.0005604044417850673,
-0.03525602072477341,
0.009365787729620934,
0.024888310581445694,
-0.007492630276829004,
-0.008225920610129833,
0.005815500859171152,
-0.012611142359673977,
0.01707622781395912,
-0.004690153989940882,
-0.043939340859651566,
-0.01309032179415226,
-0.016045264899730682,
0.019268838688731194,
-0.01649540290236473,
-0.012799910269677639,
0.017482804134488106,
-0.015115945599973202,
0.014157585799694061,
0.019007466733455658,
0.012683745473623276,
0.01203031837940216,
-0.0011680008610710502,
-0.002733503235504031,
0.02907024323940277,
0.00023550599871668965,
-0.000769138103350997,
0.02712448313832283,
-0.01175442710518837,
0.00031831880914978683,
0.022361725568771362,
0.03043518029153347,
5.714650615118444e-05,
0.0129088144749403,
-0.01229894906282425,
-0.01492717768996954,
-0.006980779115110636,
0.0020437745843082666,
-0.027589142322540283,
0.0022724741138517857,
-0.027603663504123688,
-0.05485882982611656,
0.009075376205146313,

-0.00029200021526776254,
-0.007405506446957588,
-0.011986756697297096,
-0.0035194307565689087,
-0.0382763035595417,
0.008327565155923367,
0.012429635040462017,
-0.004907963331788778,
0.0007146858260966837,
-0.03119025193154812,
-0.01908007077872753,
-0.009830446913838387,
0.013221007771790028,
0.016437320038676262,
-0.006022419314831495,
0.01012811902910471,
-0.03438478335738182,
-0.01786033995449543,
-0.005212896037846804,
0.008102495223283768,
-0.000990123488008976,
-0.0315968282520771,
0.022913508117198944,
-0.014041421003639698,
-0.00776126142591238,
-0.002813366474583745,
-0.008661538362503052,
0.004838990513235331,
-0.007695918437093496,
0.009126197546720505,
-0.025672422721982002,
-0.006276529747992754,
-0.003713643876835704,
0.016204990446567535,
-0.012277168221771717,
-0.009990173391997814,
0.0437360517680645,
0.014114024117588997,
0.044026464223861694,
-0.007434547878801823,
0.013387993909418583,
0.0009592671995051205,
-0.008066194131970406,
0.011485795490443707,
-0.00914071872830391,
-0.001017349655739963,
-0.0064943390898406506,
0.003561177523806691,

0.0035720679443329573,
-0.021011309698224068,
-0.008951950818300247,
0.011834289878606796,
0.028605584055185318,
0.014709369279444218,
0.03728890419006348,
-0.017598967999219894,
-0.003680972382426262,
-0.021635696291923523,
0.008559894748032093,
-0.03525602072477341,
0.0032580599654465914,
-0.013917996548116207,
-0.010273325257003307,
0.0016145093832165003,
-0.02584666945040226,
0.0006021512090228498,
-0.004120220430195332,
0.006007898598909378,
0.0064181056804955006,
-0.011064697988331318,
-0.018963905051350594,
0.024612419307231903,
-0.01619047112762928,
-0.0021563093177974224,
0.00041406400850974023,
0.006951737683266401,
0.00844372995197773,
0.002299700165167451,
-0.013003198429942131,
0.02286994643509388,
0.027226127684116364,
-0.01710526831448078,
0.006327352020889521,
0.02484474889934063,
0.0065923528745770454,
-0.021984189748764038,
0.003523060819134116,
0.009496472775936127,
0.03203244507312775,
-0.026006396859884262,
0.031335458159446716,
-0.0324971042573452,
0.009692501276731491,
0.013402515091001987,
0.0060442001558840275,
-0.020314320921897888,

0.032003406435251236,
0.00448686582967639,
0.025948313996195793,
0.02648557722568512,
-0.014375395141541958,
-0.002455796580761671,
-0.02615160308778286,
0.014760191552340984,
-0.012589361518621445,
-0.022245561704039574,
0.014404436573386192,
-0.013765529729425907,
-0.005554129835218191,
0.02787955477833748,
0.013446076773107052,
0.016872938722372055,
0.00268449611030519,
0.01239333301782608,
0.03121929243206978,
0.014673067256808281,
0.02472858317196369,
-0.003697308013215661,
0.008117016404867172,
-0.031335458159446716,
-0.01585649698972702,
-0.0211129542440176,
-0.014556902460753918,
-0.01858636923134327,
0.001079062232747674,
-0.04303906112909317,
0.035488348454236984,
0.005931665655225515,
0.013976078480482101,
0.006381804123520851,
0.01680033467710018,
-0.0011779838241636753,
-0.012596621178090572,
-0.008872087113559246,
0.018368560820817947,
0.011159081943333149,
-0.012698265723884106,
-0.007075163070112467,
-0.016698691993951797,
0.023218441754579544,
-0.020793501287698746,
0.017540886998176575,
-0.01393977738916874,
-0.0025574408937245607,

-0.00406213803216815,
0.007118724752217531,
-0.01423018891364336,
-0.013584022410213947,
0.027923116460442543,
-0.015711290761828423,
-0.005064059514552355,
-0.004878921899944544,
-0.001610879204235971,
0.025701463222503662,
-0.01307580154389143,
-0.00696988869458437,
0.05895363911986351,
-0.0032344639766961336,
0.020009389147162437,
-0.01588553749024868,
0.008487291634082794,
-0.019820621237158775,
-0.014005119912326336,
-9.704980038804933e-05,
-0.015130466781556606,
-0.014172106981277466,
0.0017497324151918292,
0.009699761867523193,
0.013032239861786366,
0.0035049100406467915,
0.027995718643069267,
-0.0023414469324052334,
-0.025469133630394936,
-0.02016911469399929,
0.007082423195242882,
0.010273325257003307,
0.015696769580245018,
0.017744174227118492,
0.021417886018753052,
0.005837281700223684,
-0.011064697988331318,
0.008995512500405312,
-0.03392012417316437,
-0.020967748016119003,
-0.002624598564580083,
-0.01597266085445881,
-0.011747166514396667,
-0.001030055107548833,
-0.020183635875582695,
0.01916719414293766,
0.045449480414390564,
0.027865033596754074,

-0.01022250298410654,
0.0017515475628897548,
-0.02420584298670292,
-0.026935715228319168,
-0.01889130286872387,
-0.013279089704155922,
0.021417886018753052,
0.004770017694681883,
0.04158700257539749,
0.025948313996195793,
0.010585518553853035,
-0.0028405925258994102,
0.01585649698972702,
-0.020851584151387215,
-0.02514968067407608,
0.005695705767720938,
0.02472858317196369,
0.03592396900057793,
-0.012291688472032547,
0.048063188791275024,
0.01393977738916874,
-0.010433051735162735,
0.0024576117284595966,
0.03171299397945404,
0.02326200343668461,
0.01685841754078865,
-0.00531454011797905,
0.002708092099055648,
0.014644026756286621,
0.008414688520133495,
-0.010752505622804165,
0.012023057788610458,
-0.03194532170891762,
-0.027777910232543945,
0.031016003340482712,
0.017090747132897377,
-0.02612256072461605,
-0.015290193259716034,
0.031451623886823654,
0.03290368244051933,
0.012168264016509056,
0.011093739420175552,
-0.015232110396027565,
-0.01833951845765114,
-0.019762538373470306,
0.001554611837491393,
-0.015406358055770397,
-0.01362032350152731,

-0.007950029335916042,
0.01743924245238304,
-0.031451623886823654,
-0.00855263415724039,
-0.03264231234788895,
-0.0008517240057699382,
-0.028358733281493187,
-0.019065549597144127,
-0.020459527149796486,
0.007674137596040964,
0.021374324336647987,
0.015043342486023903,
0.03243902325630188,
-0.009438390843570232,
0.014353614300489426,
0.01559512596577406,
-0.03380395844578743,
-0.009387568570673466,
0.011565659195184708,
-0.025933794677257538,
-0.0068646143190562725,
-0.03034805692732334,
0.004505016375333071,
-0.007311122491955757,
-0.0294623002409935,
0.02748749777674675,
0.00015552925469819456,
0.018354039639234543,
-0.030173810198903084,
0.00908263586461544,
0.003477683989331126,
0.016263073310256004,
0.02244884893298149,
0.03421053662896156,
-0.014389915391802788,
-0.007833864539861679,
-0.008937429636716843,
-0.019370481371879578,
-0.005833651404827833,
-0.01858636923134327,
-0.0212581604719162,
0.0014520600670948625,
0.004131110850721598,
0.005234676878899336,
-0.01562416646629572,
0.014999780803918839,
0.01630663499236107,
-0.03229381889104843,

-0.012422374449670315,
0.02286994643509388,
0.030057644471526146,
-0.0022361725568771362,
0.021650215610861778,
-0.018078148365020752,
-0.025890231132507324,
-0.021606653928756714,
-0.0022960701026022434,
-0.02523680403828621,
0.029476819559931755,
0.05906980484724045,
-0.0017070781905204058,
0.018963905051350594,
-0.007387355901300907,
-0.030609427019953728,
-0.01585649698972702,
-0.001067264238372445,
0.027298729866743088,
0.022971590980887413,
-0.012327990494668484,
-0.02044500596821308,
-0.022971590980887413,
-0.017816778272390366,
-0.013387993909418583,
0.012168264016509056,
-0.03130641579627991,
0.009903050027787685,
-0.022884467616677284,
-0.005891733802855015,
0.0034758688416332006,
-0.023973513394594193,
-0.030899839475750923,
-0.03720178082585335,
-0.012799910269677639,
0.006374543998390436,
0.01894938386976719,
0.025338448584079742,
-0.003891521133482456,
-0.0116963442414999,
-0.008828525431454182,
-0.021679257974028587,
-0.015203069895505905,
-0.01019346248358488,
0.017961984500288963,
-0.020459527149796486,
0.03116120956838131,
0.023494333028793335,

0.07800467312335968,
-0.025977356359362602,
-0.008581675589084625,
0.014513340778648853,
0.015290193259716034,
-0.011391411535441875,
-0.01562416646629572,
-0.016030743718147278,
-0.01844116300344467,
-0.025991875678300858,
0.0020891514141112566,
0.02909928560256958,
0.01828143745660782,
-0.0049478947184979916,
-0.011827029287815094,
-0.011681823991239071,
-0.019210755825042725,
0.005365361925214529,
-0.007645096629858017,
-0.05183854699134827,
-0.028532981872558594,
0.012683745473623276,
-0.025077078491449356,
0.020052950829267502,
-0.024699542671442032,
-0.02573050558567047,
0.030086684972047806,
-0.020851584151387215,
0.00943113025277853,
-8.162166341207922e-05,
0.022652138024568558,
-0.01910911127924919,
0.026311328634619713,
-0.0018767877481877804,
0.03307792916893959,
-0.00924962293356657,
0.016393758356571198,
-0.0294623002409935,
-0.008843045681715012,
0.010846888646483421,
0.00390604161657393,
0.018876781687140465,
-0.02153405174612999,
0.009133458137512207,
-0.014825534075498581,
0.02879435196518898,
-0.0022016861476004124,
-0.013228267431259155,

-0.010215243324637413,
-0.03255518898367882,
0.0012043024180456996,
-0.04486865922808647,
-0.01652444340288639,
0.005140292923897505,
0.007783042266964912,
0.024278445169329643,
0.02250693179666996,
0.02186802588403225,
-0.007979070767760277,
-0.0176134891808033,
0.013482377864420414,
-0.01073798444122076,
0.038828086107969284,
0.00034713311470113695,
0.030580386519432068,
0.0352269783616066,
0.0026554549112915993,
-0.0076087950728833675,
0.0010336852865293622,
-0.017700612545013428,
-0.01649540290236473,
0.01473115012049675,
0.030580386519432068,
-0.015304713509976864,
-0.029476819559931755,
0.008741402067244053,
0.00335607398301363,
-0.024685021489858627,
-0.0271825660020113,
0.0315968282520771,
-0.002568331314250827,
0.015653207898139954,
-0.010861409828066826,
-0.00511125149205327,
-0.007703179027885199,
0.00975058414041996,
-0.014694848097860813,
0.0027879553381353617,
0.010215243324637413,
-0.0007092406158335507,
0.007543452549725771,
0.01995130628347397,
-0.0019820621237158775,
0.02773434855043888,
0.012995937839150429,
-0.009837707504630089,

-0.0031055936124175787,
0.015130466781556606,
-0.020125553011894226,
-0.005565020255744457,
0.0031963472720235586,
0.04748236760497093,
-0.011652782559394836,
0.00511125149205327,
-0.0024920981377363205,
-4.157656439929269e-05,
-0.003826178377494216,
-0.004134741146117449,
0.0014611354563385248,
0.03551739081740379,
-0.03708561509847641,
0.024002553895115852,
-0.019660893827676773,
-0.00605146074667573,
0.013845393434166908,
0.0008440099190920591,
-0.01858636923134327,
-0.030957922339439392,
-0.022855427116155624,
-0.003205422777682543,
0.014368134550750256,
-0.005652144085615873,
0.0076305759139359,
0.0005998823326081038,
-0.010454832576215267,
-0.005732007324695587,
0.0006302848341874778,
-0.007332903798669577,
0.04347468167543411,
-0.049253880977630615,
-0.004526797216385603,
-0.011035656556487083,
-0.016960062086582184,
-0.005143922753632069,
0.009895789436995983,
0.01743924245238304,
0.0070679024793207645,
0.02284090593457222,
-0.014999780803918839,
0.0364467091858387,
-0.01057099737226963,
-0.018354039639234543,
-0.03435574099421501,
0.002651824848726392,

-0.019573770463466644,
-0.01336621306836605,
-0.0022525081876665354,
-0.0075144111178815365,
-0.0312773734331131,
-0.003116484032943845,
0.014825534075498581,
0.007659617345780134,
-0.009053594432771206,
-0.012524018995463848,
-0.001582745579071343,
0.007176807150244713,
0.006690367124974728,
-0.018818698823451996,
-0.01652444340288639,
0.04213878512382507,
-0.026558179408311844,
0.008450989611446857,
0.0005359009373933077,
-0.00794276874512434,
-0.025265846401453018,
-0.0431261844933033,
-0.010367709212005138,
-0.027923116460442543,
-0.03760835900902748,
0.0005277331219986081,
-0.0005050446488894522,
-0.020677335560321808,
-0.00845825020223856,
0.007358314469456673,
-0.02754558064043522,
-0.03955411911010742,
-0.028474899008870125,
0.024917351081967354,
-0.000542707450222224,
0.03130641579627991,
0.02976723201572895,
0.032177653163671494,
0.06598161160945892,
0.021359805017709732,
-0.014883616007864475,
0.014614985324442387,
-0.013017718680202961,
-0.018121710047125816,
-0.009903050027787685,
-0.03136449679732323,
-0.015551564283668995,
0.030202850699424744,

0.006000638473778963,
-0.016568005084991455,
-0.0023850088473409414,
-0.019457604736089706,
-0.023102276027202606,
-0.01764252968132496,
-0.0069662583991885185,
0.007155026309192181,
0.009271403774619102,
0.0020891514141112566,
0.002539290115237236,
0.008610716089606285,
0.005343581084161997,
0.019428564235568047,
-0.01619047112762928,
-0.0010155345080420375,
0.0017814962193369865,
0.0073365336284041405,
0.01377279032021761,
-0.006940847262740135,
-0.041993577033281326,
-0.00975058414041996,
0.023610496893525124,
0.032700393348932266,
0.013554980978369713,
0.019428564235568047,
0.003216313198208809,
-0.00916975922882557,
-0.009409349411725998,
-0.002484837779775262,
-0.0081242760643363,
-0.003428676864132285,
-0.009997433982789516,
0.0030602167826145887,
-0.007703179027885199,
0.016117867082357407,
-0.0067883809097111225,
-0.012422374449670315,
-0.03760835900902748,
0.02211487479507923,
0.005917144939303398,
0.0201981570571661,
-0.02554173767566681,
0.005176594480872154,
-0.006080501712858677,
-0.0244526918977499,
0.01022250298410654,
-0.01456416305154562,

0.009990173391997814,
0.02615160308778286,
-0.0008049858151935041,
-0.005710226483643055,
-0.013032239861786366,
0.030231891199946404,
-0.021969670429825783,
-0.020909665152430534,
0.02320392057299614,
0.021388845518231392,
0.01944308541715145,
-0.011355110444128513,
0.007376465480774641,
-0.0268776323646307,
-0.0028151816222816706,
-0.008770443499088287,
-0.016059784218668938,
-0.03203244507312775,
0.01594362035393715,
-0.019355962052941322,
-0.043416596949100494,
-0.038043975830078125,
-0.008814005181193352,
0.023276524618268013,
0.015566084533929825,
-0.020343361422419548,
-0.011311548762023449,
-0.01855732873082161,
0.01562416646629572,
0.0068718744441866875,
-0.019907744601368904,
-0.02120007760822773,
0.004421522840857506,
0.01054195687174797,
-0.020967748016119003,
-0.005485157016664743,
0.19968730211257935,
0.00211456255055964,
0.023160358890891075,
0.021417886018753052,
-0.007681398186832666,
0.019007466733455658,
0.0009574521682225168,
0.007292971946299076,
-0.020575691014528275,
0.01661156862974167,
0.0174682829529047,
0.0010028290562331676,

0.0025120640639215708,
-0.008980992250144482,
0.006116803269833326,
0.0349365659058094,
-0.023218441754579544,
-0.03113216906785965,
-0.025077078491449356,
-0.01492717768996954,
-0.01619047112762928,
1.1599464414757676e-05,
-0.01096305437386036,
-0.011224424466490746,
0.001359491259790957,
-0.01861541159451008,
-0.01395429763942957,
0.0025429201778024435,
0.01697458326816559,
0.01783129759132862,
-0.01186333131045103,
0.017729654908180237,
0.009191541001200676,
-0.025977356359362602,
-0.01630663499236107,
-0.017932942137122154,
0.03505273163318634,
-0.002192610874772072,
0.01858636923134327,
0.013801831752061844,
0.005227416288107634,
-0.0449557825922966,
-0.002254323335364461,
-0.019094590097665787,
-0.011362370103597641,
0.023029673844575882,
-0.009924830868840218,
-0.02189706638455391,
-0.013417035341262817,
0.0092641431838274,
-0.01359128300100565,
0.005833651404827833,
0.04974757879972458,
0.03304888680577278,
0.021461447700858116,
-0.025164201855659485,
-0.0030402508564293385,
0.0013413404813036323,
-0.010273325257003307,
0.0021181926131248474,

-0.03180011734366417,
0.013504158705472946,
-0.005234676878899336,
0.008254962041974068,
-0.024467213079333305,
-0.003087442833930254,
-0.012473196722567081,
0.012625662609934807,
0.0174682829529047,
-0.042835772037506104,
0.003826178377494216,
-0.03996069356799126,
0.03354258835315704,
0.0016934651648625731,
-0.01517402846366167,
-0.03203244507312775,
0.01908007077872753,
0.006461667362600565,
0.015115945599973202,
0.06383256614208221,
-0.01658252626657486,
-0.027690786868333817,
0.0046502226032316685,
-0.01187785156071186,
0.002163569675758481,
-0.02286994643509388,
0.002617338439449668,
-0.009692501276731491,
-0.015101425349712372,
-0.01719239167869091,
-0.011551138013601303,
-0.006330982316285372,
0.001923979609273374,
0.01057099737226963,
0.02080802246928215,
-0.013017718680202961,
-0.00838564708828926,
0.015740331262350082,
0.002421310171484947,
0.00792824849486351,
-0.028228048235177994,
0.03551739081740379,
0.02423488348722458,
-0.0041093300096690655,
-0.03365875408053398,
-0.005285498686134815,
0.010345928370952606,
-0.002348707290366292,

-0.015958141535520554,
-0.027603663504123688,
0.016059784218668938,
-0.01195045467466116,
0.0009465616894885898,
-0.002953127259388566,
0.008313043974339962,
0.0012260832590982318,
0.006076871417462826,
-0.03034805692732334,
0.005118512082844973,
0.003461348358541727,
0.0012569394893944263,
-0.023131318390369415,
0.0008798576891422272,
0.008951950818300247,
-0.013475118204951286,
-0.013417035341262817,
-0.01473115012049675,
0.002539290115237236,
-0.022681178525090218,
0.010011954233050346,
0.015929099172353745,
-0.0040185763500630856,
0.010019214823842049,
-0.024351047351956367,
-0.005089470651000738,
0.021969670429825783,
0.0067702303640544415,
-0.003746315138414502,
0.017424721270799637,
-0.010273325257003307,
0.010512915439903736,
0.010011954233050346,
0.0023069605231285095,
-0.030667509883642197,
-0.017381159588694572,
-0.022986112162470818,
-0.00645440723747015,
-0.00531454011797905,
-0.005721116904169321,
0.0076160551980137825,
-0.012879773043096066,
-0.015449919737875462,
-0.01229894906282425,
-0.02378474548459053,
0.027356812730431557,
-0.022289123386144638,

-0.023363647982478142,
-0.017961984500288963,
-0.02512064017355442,
0.004043987486511469,
-0.055178284645080566,
0.006559681612998247,
0.01950116828083992,
-0.015028822235763073,
-0.020996788516640663,
-0.028939558193087578,
-0.18760617077350616,
0.026078999042510986,
0.02041596546769142,
-0.026863113045692444,
0.05047360807657242,
0.025323929265141487,
0.04376509040594101,
-0.014462518505752087,
0.003036620793864131,
-0.010244283825159073,
-0.0077104391530156136,
-0.0026863112580031157,
-0.03362971171736717,
-0.029694629833102226,
-0.014498820528388023,
0.014266490936279297,
-0.0011552952928468585,
0.01353320013731718,
0.025222284719347954,
0.016814855858683586,
0.022361725568771362,
-0.022608576342463493,
0.012857992202043533,
0.01682937704026699,
-0.018049107864499092,
-0.008755922317504883,
-0.027327772229909897,
0.009663459844887257,
0.001989322481676936,
-0.009314965456724167,
-0.02038692496716976,
-0.003165491158142686,
0.019239796325564384,
-0.025498175993561745,
0.0312773734331131,
-0.0036501160357147455,
0.014375395141541958,
-0.020328842103481293,

-0.01239333301782608,
0.011986756697297096,
0.0038515895139425993,
0.0388861708343029,
-0.003232648828998208,
-0.0024249404668807983,
-0.0054125539027154446,
0.014985260553658009,
0.024699542671442032,
-0.02375570312142372,
-0.0069481078535318375,
-0.006218447349965572,
0.008523592725396156,
-0.011202643625438213,
-0.01072346419095993,
-0.005256457719951868,
0.015362796373665333,
0.02278282307088375,
-0.027719827368855476,
0.013896215707063675,
0.003198162419721484,
-0.00101190444547683,
0.005695705767720938,
0.004363440442830324,
-0.006312831304967403,
-0.006806531921029091,
0.004428783431649208,
-0.008705100044608116,
-0.03508177399635315,
0.010338667780160904,
0.007155026309192181,
0.0006030587246641517,
0.009656200185418129,
0.023044195026159286,
0.004142001271247864,
0.018688013777136803,
0.013671145774424076,
0.01362032350152731,
-0.012654704041779041,
0.0185282863676548,
0.013663886114954948,
-0.01022250298410654,
-0.015958141535520554,
0.05866323038935661,
-0.0221729576587677,
0.011848810128867626,
0.01813623122870922,
0.0053762528114020824,

0.009736062958836555,
-0.006566941738128662,
0.00838564708828926,
-0.0014801938086748123,
0.01864445209503174,
-0.011231685057282448,
-0.0059425560757517815,
-0.006614133715629578,
0.004755496978759766,
0.008087974973022938,
0.0013077616458758712,
0.003935082815587521,
0.01140593271702528,
0.0035756980068981647,
-0.0007060642237775028,
-0.0029004900716245174,
-0.004014946054667234,
0.010418531484901905,
0.021998710930347443,
-0.012879773043096066,
-0.023770224303007126,
0.020909665152430534,
0.03952507674694061,
-0.03354258835315704,
-0.012052099220454693,
0.02314583770930767,
6.307386502157897e-05,
0.032874640077352524,
0.02016911469399929,
0.02706640027463436,
-0.008893867954611778,
-0.014339094050228596,
0.014418956823647022,
0.0077104391530156136,
0.05683363229036331,
-0.009177019819617271,
0.004926113877445459,
-0.004004055634140968,
-0.0233926884829998,
0.0025592560414224863,
-0.0805022120475769,
-0.024917351081967354,
-0.004221864975988865,
0.0074309175834059715,
-0.023000631481409073,
0.0011108260368928313,
0.004290837794542313,
0.026195164769887924,

-0.0028042912017554045,
-0.007020710501819849,
0.002767989644780755,
0.009910310618579388,
0.012494977563619614,
0.018005546182394028,
0.015987182036042213,
-0.014687588438391685,
-0.006280160043388605,
0.002394084120169282,
-0.019965827465057373,
0.03714369609951973,
0.014571423642337322,
-0.010658121667802334,
-0.007347424048930407,
-0.01186333131045103,
-0.0005490602343343198,
0.016393758356571198,
-0.036011092364788055,
0.011122780852019787,
0.015696769580245018,
-0.023058714345097542,
-0.003700938308611512,
-0.027589142322540283,
-0.013083062134683132,
-0.00906811561435461,
-0.01958829164505005,
-0.0057755690068006516,
0.0018767877481877804,
-0.007376465480774641,
0.03627246245741844,
-0.042051661759614944,
0.02851846069097519,
0.007420027162879705,
0.0209532268345356,
-0.0025247696321457624,
-0.0050313882529735565,
0.02120007760822773,
-0.005750157870352268,
0.03380395844578743,
0.003495834767818451,
-0.008632497861981392,
-0.03104504570364952,
-0.007717699743807316,
0.002058295300230384,
0.004243645817041397,
0.026296809315681458,
-0.006799271330237389,

0.007169547025114298,
0.015682248398661613,
0.0016571636078879237,
-0.02907024323940277,
0.00010635206126607955,
-0.02180994302034378,
-0.008175098337233067,
0.012487716972827911,
0.015406358055770397,
0.00619666650891304,
-0.0031237443909049034,
-0.025599820539355278,
0.0021381585393100977,
-0.008690579794347286,
-0.018295956775546074,
0.0458560585975647,
-0.023378167301416397,
0.008777703158557415,
-0.013903475366532803,
-0.02639845386147499,
0.004621181171387434,
-0.03243902325630188,
-0.0004351642564870417,
-0.006566941738128662,
-0.015464439988136292,
-0.011180862784385681,
-0.0021073021925985813,
0.0018786027794703841,
0.018121710047125816,
-0.0031909020617604256,
0.00331069715321064,
-0.0013876250013709068,
0.028663666918873787,
-0.051954712718725204,
0.019573770463466644,
0.0009701576782390475,
0.016103345900774002,
-0.02025623805820942,
0.0009801405249163508,
0.030754633247852325,
-0.0017860339721664786,
-0.021025830879807472,
0.0021054872777312994,
0.01955924928188324,
-0.06278707832098007,
-0.013983339071273804,
-0.0871816873550415,
0.04042535275220871,

0.0014783786609768867,
-0.026950236409902573,
0.015987182036042213,
-0.04460728541016579,
-0.00505679938942194,
-0.006908176001161337,
0.009409349411725998,
0.020880624651908875,
-0.019268838688731194,
0.00977962464094162,
0.004407002590596676,
-0.0032344639766961336,
-0.013235528022050858,
-0.01125346589833498,
0.032177653163671494,
-0.014651286415755749,
0.024888310581445694,
0.030754633247852325,
-0.013649364933371544,
-0.034007247537374496,
0.005917144939303398,
0.02949134074151516,
-0.019312400370836258,
-0.004112960305064917,
0.009656200185418129,
0.01664060913026333,
-0.0062220776453614235,
-0.013308131136000156,
-0.007303862366825342,
-0.026892153546214104,
0.002377748489379883,
0.010157160460948944,
0.004152892157435417,
-0.0021563093177974224,
0.012625662609934807,
0.010636340826749802,
0.034675195813179016,
0.010360448621213436,
-0.019791578873991966,
-0.023058714345097542,
0.03516889736056328,
-0.029171887785196304,
-0.014999780803918839,
0.0016508108237758279,
0.004998716991394758,
-0.000693358713760972,
0.018426643684506416,
0.0013731044018641114,

0.034704238176345825,
0.01563868671655655,
-0.016843898221850395,
-0.01998034678399563,
0.013642105273902416,
-0.025686943903565407,
-0.001969356555491686,
0.004806318785995245,
0.015014301985502243,
0.019094590097665787,
0.042661525309085846,
-0.005027757957577705,
0.013032239861786366,
-0.015711290761828423,
0.028620105236768723,
0.0011607405031099916,
-0.0030166548676788807,
0.004142001271247864,
0.002862373599782586,
-0.0221729576587677,
-0.007913727313280106,
-0.0002359597710892558,
0.008893867954611778,
-0.004105700179934502,
0.011674563400447369,
0.023770224303007126,
0.0012542168842628598,
-0.014542382210493088,
0.010854149237275124,
0.019965827465057373,
0.009293184615671635,
-0.021737340837717056,
-0.012959636747837067,
0.007884686812758446,
0.034094370901584625,
0.014310052618384361,
-0.010186201892793179,
-0.005162073764950037,
0.011101000010967255,
-0.011326069012284279,
-0.012371552176773548,
0.0022869945969432592,
0.011231685057282448,
0.03987357020378113,
0.013402515091001987,
0.01794746331870556,
-0.023683100938796997,
-0.011660043150186539,

0.023973513394594193,
-0.006414475385099649,
0.014781972393393517,
0.0072348895482718945,
0.015508001670241356,
-0.03113216906785965,
-0.024409130215644836,
0.014506080187857151,
-0.01844116300344467,
-0.012828950770199299,
0.0033288479316979647,
-0.005713856779038906,
0.010360448621213436,
-0.01725047454237938,
-0.0011062882840633392,
0.02047404833137989,
-0.009358527138829231,
0.013961558230221272,
-0.01193593442440033,
-0.012015797197818756,
-0.02205679379403591,
0.05761774629354477,
0.007543452549725771,
0.00993209145963192,
0.06313557177782059,
-0.012734566815197468,
0.03429765999317169,
0.02790859527885914,
-0.013170185498893261,
0.021374324336647987,
0.001773328403942287,
0.012516758404672146,
-0.001728859031572938,
0.003985905088484287,
-0.022521452978253365,
-0.025774067267775536,
-0.00914071872830391,
-0.0014620430301874876,
-0.035372182726860046,
-0.012066619470715523,
0.004737345967441797,
0.1123894527554512,
0.020764458924531937,
-0.029563944786787033,
0.011362370103597641,
-0.02988339774310589,
0.018078148365020752,
0.008058933541178703,

```

        0.0069481078535318375,
        0.009198800660669804,
        -0.04028014838695526,
        -0.02968010865151882,
        -0.014760191552340984,
        -0.03789876773953438,
        -0.023639539256691933,
        -0.03850863501429558,
        0.011442233808338642,
        -0.005350841674953699,
        0.028605584055185318,
        0.004711935296654701,
        -0.024380089715123177,
        0.02284090593457222,
        0.029288053512573242,
        0.02047404833137989,
        -8.196198905352503e-05,
        -0.0247576255351305,
        -0.005855432245880365,
        0.028503939509391785,
        -0.007979070767760277,
        -0.0004791798419319093,
        -0.013337172567844391,
        0.012306209653615952,
        -0.017700612545013428,
        0.0008149687200784683,
        0.004675633739680052,
        0.004145631566643715,
        0.012429635040462017,
        -0.011006616055965424,
        -0.01552252285182476,
        0.021650215610861778,
        -0.0073292735032737255,
        0.0031745664309710264,
        -0.0036101844161748886,
        -0.015566084533929825,
        -0.03908945992588997,
        0.04022206366062164,
        -0.004436043556779623,
        0.020052950829267502,
        -0.016538964584469795,
        -0.039350830018520355
    ],
}
},
{
  "_index": "github_issues",
  "_id": "BWLQUZMBKaR7I-arfsTD",

```

```
"_score": 1.0,
"_source": {
    "_type": "issue",
    "_repo": "langchain",
    "_issueNumber": "28273",
    "_title": "api-docs: add standard tests package to build 2",
    "_createdAt": "2024-11-21T23:40:37Z",
    "_closedAt": "2024-11-21T23:40:48Z",
    "_state": "closed",
    "_body": "None",
    "GitHub_Issue_vector": [
        0.0007092204759828746,
        0.0012419006088748574,
        0.009564644657075405,
        -0.001148386625573039,
        -0.02939310297369957,
        0.016500409692525864,
        -0.005953431595116854,
        -0.028092646971344948,
        -0.014990203082561493,
        -0.0016675202641636133,
        0.02227555401623249,
        -0.00402721855789423,
        0.0061806621961295605,
        0.00505849439650774,
        -0.005537425633519888,
        -0.0011143020819872618,
        0.015591489151120186,
        -0.024415012449026108,
        0.019968291744589806,
        0.02223360352218151,
        0.013731696642935276,
        0.012095639482140541,
        -0.0061457036063075066,
        0.00987227912992239,
        -0.02644260600209236,
        0.02571546845138073,
        0.009410827420651913,
        -0.024415012449026108,
        0.005977902561426163,
        0.007725827395915985,
        0.035098329186439514,
        -0.027001941576600075,
        -0.005313691217452288,
        -0.02526800148189068,
        -0.006065298803150654,
        -0.004604033660143614,
        0.016542360186576843,
```

0.004170548636466265,
0.014137215912342072,
-0.01638854295015335,
0.012487174943089485,
0.00960659421980381,
0.017395347356796265,
-0.019408956170082092,
-0.01876571960747242,
-0.004705413710325956,
0.030036339536309242,
-0.011256635189056396,
-0.0033857303205877542,
0.015102069824934006,
0.012165556661784649,
-0.004922156222164631,
0.0017610342474654317,
0.02121281437575817,
0.012067672796547413,
0.01005406305193901,
-0.00044353591511026025,
0.022135719656944275,
0.007348275743424892,
-0.01553555577993393,
-0.005869531538337469,
-0.004174044355750084,
-0.029085468500852585,
-0.0021901498548686504,
-0.013584871776401997,
-0.012179539538919926,
0.0022286041639745235,
0.009634560905396938,
0.006460329983383417,
-0.031658414751291275,
0.0318821482360363,
0.01234734058380127,
0.0005790001014247537,
-0.012941635213792324,
0.043348535895347595,
-0.0010391413234174252,
0.002483801217749715,
-0.02510019950568676,
-0.008893441408872604,
-0.012780825607478619,
0.02686210721731186,
-0.0009762159897945821,
-0.017758915200829506,
0.023981528356671333,
0.03246945142745972,

0.01255009975284338,
0.016010990366339684,
0.01933903992176056,
0.003175979247316718,
-0.014752484858036041,
0.034035589545965195,
0.008264188654720783,
0.01676609367132187,
0.02031787671148777,
-0.0008647858048789203,
0.04849442467093468,
-0.01387852244079113,
0.04745965451002121,
0.035014428198337555,
-0.05956927686929703,
-0.007026657462120056,
0.007474126294255257,
-0.02852613292634487,
-0.0026446101255714893,
-0.002160435076802969,
-0.02174418419599533,
0.0029417574405670166,
-0.0002436170179862529,
0.02157638408243656,
0.014976219274103642,
-0.0274354275316,
0.03834247961640358,
0.014514767564833164,
-0.055989526212215424,
0.02599513716995716,
-0.0007743306923657656,
-0.007572010159492493,
-0.036608535796403885,
0.0017120923148468137,
0.0038244593888521194,
0.026400655508041382,
0.013535929843783379,
0.014864352531731129,
0.0018930025398731232,
-0.0027320063672959805,
0.002665585372596979,
-0.01679406128823757,
-0.0318821482360363,
-0.008816532790660858,
-0.0073622590862214565,
0.01655634306371212,
0.013312195427715778,
-0.0020293407142162323,

-0.010851116850972176,
-0.01279480941593647,
0.0036531628575176,
-0.009054250083863735,
0.020122110843658447,
-0.014486800879240036,
-0.026316754519939423,
-0.004107623361051083,
0.03495849668979645,
-0.020877214148640633,
-0.021156881004571915,
-0.026051070541143417,
0.030567709356546402,
0.008732631802558899,
0.00449566263705492,
0.02399551123380661,
-0.03339235484600067,
0.0012480182340368629,
-0.024387046694755554,
-0.008487923070788383,
-0.0018964983755722642,
-0.011172735132277012,
0.022695055231451988,
-0.01193483080714941,
0.012738876044750214,
-0.03263724967837334,
0.008187280036509037,
-0.005436046048998833,
0.01995430886745453,
0.004306886810809374,
-0.03658057004213333,
0.007599976845085621,
0.028106629848480225,
0.018653852865099907,
-0.009543669410049915,
-0.010165930725634098,
-0.008466947823762894,
-0.00938285980373621,
0.0027320063672959805,
-0.04169849306344986,
0.029085468500852585,
-0.00505849439650774,
0.0389297790825367,
-0.0035535311326384544,
0.01961870677769184,
-0.017339413985610008,
0.0038419386837631464,
-0.028512148186564445,

-0.0021167369559407234,
0.011102817952632904,
0.054059818387031555,
-0.0005938574322499335,
0.011417444795370102,
0.011843938380479813,
0.005512954667210579,
0.02342219278216362,
0.007135028950870037,
0.0247785821557045,
0.019436923786997795,
-0.008418005891144276,
0.006222612224519253,
-0.614374577999115,
-0.0072154332883656025,
0.0009360137046314776,
-0.012487174943089485,
0.01148736197501421,
0.028749866411089897,
0.013619829900562763,
-0.0017418070929124951,
0.002413884038105607,
0.012053688988089561,
-0.02677820809185505,
0.026316754519939423,
0.02010812610387802,
-0.007935578934848309,
-0.002912042662501335,
-0.016122858971357346,
0.03274912014603615,
-0.00921505969017744,
-0.02530995011329651,
0.00993520487099886,
-0.04611724615097046,
0.02289082296192646,
0.004600537940859795,
-0.008627756498754025,
0.023408208042383194,
-0.029532937332987785,
0.01949285715818405,
0.006932269781827927,
0.0075090848840773106,
0.0026463582180440426,
-0.022834889590740204,
0.02914140187203884,
0.010522507131099701,
-0.0003640053328126669,
0.04346040263772011,

0.0074951015412807465,
-0.009033274836838245,
0.008809540420770645,
0.007002186495810747,
0.011235660873353481,
-0.024988332763314247,
0.0052018240094184875,
0.0001756664423737675,
0.007823711261153221,
-0.007907611317932606,
-0.014486800879240036,
0.02010812610387802,
0.007068607956171036,
-0.010676324367523193,
0.009760412387549877,
0.0323016494512558,
-0.015325804241001606,
-0.005981398746371269,
0.008431988768279552,
0.0021796622313559055,
-0.03269318491220474,
0.029281236231327057,
-0.02342219278216362,
0.005775143392384052,
0.00829914677888155,
0.00595692778006196,
0.0061561912298202515,
0.007627943530678749,
-0.019800491631031036,
-0.008711657486855984,
0.008655723184347153,
0.004450216423720121,
0.025939203798770905,
0.012654975056648254,
-0.04385193809866905,
-0.006243587471544743,
-0.018877588212490082,
-0.00438379542902112,
-0.007054624613374472,
0.013892506249248981,
0.02911343425512314,
0.01806654967367649,
0.00899132527410984,
-0.03470679372549057,
0.018863603472709656,
-0.0010548726422712207,
-0.011976780369877815,
-0.019926343113183975,

-0.009522694163024426,
0.020136093720793724,
0.01171109639108181,
-0.01683601178228855,
0.0038419386837631464,
-0.000533991027623415,
-0.011676137335598469,
0.009068233892321587,
0.01003308780491352,
0.015171987004578114,
-0.00807541236281395,
0.002768712816759944,
0.028386298567056656,
0.016626261174678802,
0.009250017814338207,
0.02391161024570465,
-0.025771403685212135,
-0.013214311562478542,
0.01798265054821968,
-0.008005495183169842,
0.008851490914821625,
0.03062364272773266,
-0.0046949260868132114,
-0.026190904900431633,
0.0157313235104084,
0.011431427672505379,
0.002695299917832017,
-0.01966065727174282,
0.007369250990450382,
-0.006327487528324127,
0.019143272191286087,
-0.0006310008466243744,
-0.04320869967341423,
-0.015871157869696617,
-0.027337543666362762,
0.0018667837139219046,
0.0188076701015234,
-0.020919164642691612,
0.0006344966823235154,
-0.001989138312637806,
0.014032339677214622,
0.01721356250345707,
0.00663162674754858,
-0.00991422962397337,
-0.014878335408866405,
-0.006397404707968235,
-0.0016343096503987908,
-0.0072154332883656025,

0.022904805839061737,
0.014360950328409672,
-0.009040267206728458,
0.018541986122727394,
0.01770298182964325,
0.015185970813035965,
-0.009802361950278282,
0.015969039872288704,
-0.0006851865327917039,
-0.006603659596294165,
0.00018495229596737772,
0.006467321887612343,
-0.0247785821557045,
0.01933903992176056,
-0.04099932312965393,
-0.026148954406380653,
-0.0014595171669498086,
-0.0038838889449834824,
0.0061561912298202515,
-0.018947504460811615,
-0.005589863285422325,
-0.01691991090774536,
-0.017199579626321793,
-0.0018877587281167507,
0.0036636502481997013,
0.030036339536309242,
-0.03160247951745987,
-0.0038489303551614285,
-0.0008237095316872001,
0.024149328470230103,
-0.011976780369877815,
-0.02129671536386013,
-0.012060681357979774,
-0.019898375496268272,
-0.0034381679724901915,
-0.02214970253407955,
0.022009868174791336,
-0.0074951015412807465,
-0.03364405781030655,
-0.0018755232449620962,
-0.015409705229103565,
-0.006397404707968235,
-0.011983771808445454,
0.009683502838015556,
0.0062051331624388695,
-0.010704291053116322,
0.002779200440272689,
-0.02730957604944706,

-0.005027031525969505,
-0.0033175612334161997,
0.006432363297790289,
-0.025198083370923996,
0.008215246722102165,
0.03135078027844429,
0.008753607049584389,
0.027519328519701958,
0.004222986288368702,
0.008271180093288422,
-0.003432924160733819,
-0.02607903815805912,
0.010683315806090832,
-0.011333543807268143,
0.0005894876085221767,
-0.023729827255010605,
0.001999625936150551,
-0.00033822344266809523,
0.006016356870532036,
0.007341283839195967,
0.026512522250413895,
0.001636931556276977,
0.01185792125761509,
-0.006009365431964397,
-0.013235286809504032,
0.00704064080491662,
-0.016584310680627823,
0.005883514881134033,
-0.029700737446546555,
0.009124167263507843,
-0.011249643750488758,
0.01052949856966734,
-0.019856425002217293,
-0.02582733705639839,
-0.010103004984557629,
0.005054998677223921,
-0.005677259527146816,
-0.004226482007652521,
-0.0165982935577631,
-0.014836385846138,
0.0044152578338980675,
-0.013507962226867676,
-0.002592172473669052,
0.000822398578748107,
-0.0011894628405570984,
0.013815597631037235,
-0.0165982935577631,
0.009641553275287151,

0.036272935569286346,
0.022289536893367767,
0.012291407212615013,
-0.0022513270378112793,
0.018220368772745132,
0.006691055838018656,
-0.008418005891144276,
0.022834889590740204,
0.015409705229103565,
0.02722567692399025,
-0.015102069824934006,
0.029980406165122986,
-0.023855676874518394,
0.022834889590740204,
0.01434696651995182,
0.01623472571372986,
-0.014360950328409672,
0.03551783040165901,
-0.0034556472674012184,
0.027687128633260727,
0.005072477739304304,
0.005792622920125723,
-0.010354706086218357,
0.00783070269972086,
-0.011962797492742538,
-0.0037999884225428104,
-0.011990764178335667,
0.014584684744477272,
-0.0174932312220335,
-0.024456962943077087,
0.03241351619362831,
-0.0007227669120766222,
0.02265310473740101,
0.014990203082561493,
-0.013773647136986256,
0.02661040611565113,
0.01676609367132187,
0.03070754185318947,
-0.01279480941593647,
-0.004236969631165266,
-0.01843011938035488,
0.017395347356796265,
-0.011333543807268143,
0.0029330176766961813,
-0.007837695069611073,
0.0020345845259726048,
-0.017591115087270737,
0.05179450660943985,

0.00834109727293253,
0.00584506057202816,
0.02076534554362297,
0.02616293728351593,
-0.02093314751982689,
-0.03070754185318947,
-0.025044266134500504,
0.001556526985950768,
0.00942481029778719,
-0.003925838973373175,
-0.0008674076525494456,
-0.020136093720793724,
-0.00036291289143264294,
-0.025114182382822037,
0.017884766682982445,
-0.002111493144184351,
-0.006904303096234798,
-0.01526987086981535,
-0.024289162829518318,
-0.017912732437253,
-0.001830077264457941,
0.0219119843095541,
-0.016206758096814156,
0.002392909023910761,
0.021450532600283623,
0.007404209580272436,
0.007072103675454855,
0.02686210721731186,
-0.0023544547148048878,
0.018569953739643097,
0.01708771288394928,
0.00015731323219370097,
-0.036608535796403885,
-0.0002960547572001815,
-0.0028508652467280626,
0.021492483094334602,
0.011389478109776974,
0.016360575333237648,
-0.006981211714446545,
-0.010389665141701698,
0.0009867035550996661,
0.0026760727632790804,
-0.020541612058877945,
0.016458459198474884,
0.003210937837138772,
-0.011941822245717049,
-0.027029909193515778,
0.002515263855457306,

0.008313130587339401,
0.022331487387418747,
0.047152020037174225,
0.018248334527015686,
0.0066735767759382725,
-0.016933895647525787,
0.022303519770503044,
-0.012074664235115051,
0.015563522465527058,
-0.00020691059762611985,
-0.01700381189584732,
-0.018444102257490158,
-0.008977341465651989,
-0.0004334853438194841,
0.007033649366348982,
0.036161068826913834,
0.019325057044625282,
-0.009795370511710644,
-0.017605097964406013,
-0.022499287500977516,
0.005603846628218889,
-0.001019040122628212,
-0.03179824724793434,
0.013011552393436432,
0.0071944585070014,
0.04908172786235809,
-0.0034346722532063723,
-0.024876466020941734,
0.011948813684284687,
0.026848124340176582,
-0.015088086947798729,
0.01545165479183197,
0.002712779212743044,
-0.0003980898763984442,
0.030679576098918915,
-0.02461078017950058,
0.039181482046842575,
-0.004597042221575975,
0.0005042763077653944,
-0.0075300601311028,
0.02227555401623249,
0.008397030644118786,
-0.002149947453290224,
-0.0060583073645830154,
-0.02301667258143425,
0.0150601202622056,
0.006785443983972073,
-0.020737379789352417,

0.003513328731060028,
-0.0014516515657305717,
-0.01953480765223503,
0.025561651214957237,
0.02461078017950058,
-0.038482312113046646,
-0.0278828963637352,
0.04144679382443428,
0.023212440311908722,
-0.0274354275316,
0.02159036695957184,
-0.02718372642993927,
-0.014906302094459534,
-0.02854011580348015,
0.0027075354009866714,
-0.024624764919281006,
-0.027771029621362686,
-0.026708289980888367,
-0.009984145872294903,
-0.02293277345597744,
-0.010207880288362503,
-0.03392372280359268,
0.0050794691778719425,
0.007844686508178711,
-0.012053688988089561,
-0.006603659596294165,
0.016752110794186592,
0.02375779300928116,
0.004775330424308777,
0.010347714647650719,
-0.007928586564958096,
0.006827394012361765,
0.027029909193515778,
-0.019325057044625282,
-0.022862855345010757,
0.01389949768781662,
0.0005422936519607902,
0.02358999289572239,
0.010879083536565304,
0.0006117736920714378,
-0.005254261661320925,
-0.005390599835664034,
0.035825464874506,
-0.01806654967367649,
0.015521571971476078,
-0.009543669410049915,
-0.03495849668979645,
0.01643049344420433,

0.0034311763010919094,
0.005586367566138506,
0.014892319217324257,
-0.01712966337800026,
0.010061055421829224,
-0.004971098154783249,
-0.03171434625983238,
-0.0210170466452837,
-0.02371584251523018,
-0.0052262949757277966,
-8.919878746382892e-05,
-0.010830141603946686,
0.012578066438436508,
-0.037922974675893784,
0.014263066463172436,
0.0021953934337943792,
-0.016668209806084633,
-0.005705226678401232,
0.007002186495810747,
0.008627756498754025,
0.019394973292946815,
0.012892693281173706,
0.016542360186576843,
-0.010990951210260391,
-0.008271180093288422,
0.015885140746831894,
-0.0331965871155262,
0.015843190252780914,
0.02473663166165352,
0.005991885904222727,
0.019520824775099754,
-0.01430501602590084,
-0.013983398675918579,
-0.02542181871831417,
-0.002786192111670971,
0.010277797468006611,
0.013277236372232437,
0.01301854383200407,
-0.0014769964618608356,
0.0026009120047092438,
-0.03221774846315384,
0.007551034912467003,
0.001268119434826076,
-0.009578627534210682,
-0.0019157255301252007,
-0.01823435164988041,
0.001964667346328497,
-0.017241530120372772,

-0.019059371203184128,
4.8859958042157814e-05,
-0.03837044537067413,
-0.005411575082689524,
0.01169012114405632,
0.004111119080334902,
0.022345470264554024,
-0.028582066297531128,
-0.004443224985152483,
0.010676324367523193,
0.013172361068427563,
-0.014109249226748943,
-0.02301667258143425,
0.006124728359282017,
-0.006806419230997562,
0.03356015682220459,
0.036272935569286346,
0.026540489867329597,
-0.008872466161847115,
0.016122858971357346,
-0.011480369605123997,
0.02841426432132721,
-0.009802361950278282,
-0.027519328519701958,
-0.0022583187092095613,
-0.00920107588171959,
-0.008823524229228497,
0.014990203082561493,
0.015130036510527134,
0.002431363333016634,
0.004149573389440775,
-0.015759289264678955,
0.023002689704298973,
-0.018500035628676414,
0.000616580480709672,
-0.02084924653172493,
-0.04622911289334297,
-0.024498913437128067,
0.012200514785945415,
-0.04320869967341423,
0.019814474508166313,
-0.04206206277012825,
-0.0179407000541687,
0.05288521200418472,
-0.004079656209796667,
-0.0008529872866347432,
-0.0043173739686608315,
0.032273683696985245,

-0.020122110843658447,
0.013969414867460728,
0.004593546502292156,
0.009229042567312717,
-0.020793313160538673,
0.016360575333237648,
-0.007537051569670439,
-0.005883514881134033,
0.008900432847440243,
0.013088461011648178,
0.020387794822454453,
-0.00357625400647521,
0.01054348237812519,
-0.01933903992176056,
0.04206206277012825,
-0.0327770859003067,
-0.007257383782416582,
-0.020219994708895683,
-0.021184848621487617,
-0.00035592119093053043,
-0.020513644441962242,
0.013752671889960766,
-0.02178613469004631,
0.02154841646552086,
0.01187889650464058,
-0.0027407461311668158,
0.012319373898208141,
-0.02902953512966633,
0.003527312306687236,
0.022387420758605003,
-0.016849994659423828,
0.04278919845819473,
0.038202643394470215,
0.021772151812911034,
0.0219119843095541,
0.022415386512875557,
-0.026582440361380577,
-0.0024890447966754436,
-0.0265684574842453,
-0.007551034912467003,
0.02223360352218151,
0.019604723900556564,
-0.009515702724456787,
-0.029253268614411354,
0.022709038108587265,
0.01263400074094534,
-0.0038664096500724554,
-0.03870604559779167,

0.003095574676990509,
-0.0009631065768189728,
0.011207693256437778,
-0.028064679354429245,
0.009068233892321587,
-0.024820532649755478,
0.0073133171536028385,
-0.00850190594792366,
-0.011102817952632904,
-0.010620390996336937,
-0.008935391902923584,
0.014074290171265602,
0.013724705204367638,
-0.008599789813160896,
0.036356836557388306,
0.02665235660970211,
-0.028302397578954697,
0.004978089593350887,
-0.001299582072533667,
0.004254448693245649,
0.01912928931415081,
0.017814848572015762,
0.03867807984352112,
0.0009316438809037209,
0.003050128696486354,
0.015759289264678955,
0.0023631942458450794,
-0.012878709472715855,
-0.0002991136279888451,
0.020415760576725006,
0.03915351629257202,
-0.020555594936013222,
0.00017446474521420896,
0.00728535046800971,
-0.0146126514300704,
-0.0008665337227284908,
-0.01933903992176056,
-0.012032713741064072,
-0.022415386512875557,
-0.015689373016357422,
0.003155004233121872,
0.012787817977368832,
0.023072607815265656,
-0.008355080150067806,
-0.01610887423157692,
-0.013948439620435238,
-0.005110932048410177,
-0.03196604922413826,

-0.013584871776401997,
0.010641366243362427,
-0.020136093720793724,
-0.03311268612742424,
-0.0024890447966754436,
-0.029504969716072083,
0.01531182136386633,
-0.001247144304215908,
0.010739250108599663,
-0.01585717312991619,
0.03859417885541916,
-0.007711844053119421,
0.028973601758480072,
-0.018961487337946892,
-0.024876466020941734,
-0.01455671712756157,
0.005729697644710541,
-0.012606033124029636,
-0.014151198789477348,
0.008655723184347153,
-0.019269121810793877,
-0.04751558601856232,
-0.006729510612785816,
0.013969414867460728,
0.020080160349607468,
-0.006215620320290327,
-0.028624016791582108,
0.007893628440797329,
0.009347901679575443,
0.004789313767105341,
-0.009340910241007805,
-0.02333829179406166,
0.049529194831848145,
-0.011102817952632904,
0.000983207719400525,
-0.002020600950345397,
-0.0061072492972016335,
-0.023450158536434174,
0.0007765155751258135,
0.001629939884878695,
0.003918847534805536,
-0.018248334527015686,
-0.02742144465446472,
0.003618204267695546,
0.004233473911881447,
0.0036321876104921103,
-0.02416331134736538,
-0.011312569491565228,

-0.038286544382572174,
-0.02690405771136284,
0.028582066297531128,
0.015787256881594658,
0.004380299709737301,
0.004278919659554958,
0.003674137871712446,
0.030399907380342484,
0.032525382936000824,
-0.023310324177145958,
-0.01124265231192112,
-0.02758924476802349,
-0.015913106501102448,
-0.014892319217324257,
-0.030791442841291428,
-0.01774493232369423,
0.023576010018587112,
0.029644804075360298,
0.009529685601592064,
-0.009515702724456787,
-0.014151198789477348,
-0.0060478197410702705,
0.0010671081254258752,
-0.0028648488223552704,
-0.014088273979723454,
0.0074112010188400745,
-0.0014936017105355859,
-0.012487174943089485,
0.017241530120372772,
-0.007732819300144911,
0.02150646597146988,
-0.003604220924898982,
-0.02391161024570465,
-0.004076160490512848,
-0.0018860108684748411,
0.011836946941912174,
-0.02886173315346241,
-0.00921505969017744,
0.0060478197410702705,
0.018541986122727394,
0.022457337006926537,
0.021156881004571915,
0.011697112582623959,
-0.00811037141829729,
-0.010452589951455593,
-0.027029909193515778,
-0.011557278223335743,
-0.007676885463297367,

0.01526987086981535,
0.005547913257032633,
-0.019520824775099754,
-0.02166028320789337,
0.010760225355625153,
0.0078656617552042,
0.02714177593588829,
-0.05668869614601135,
0.008047445677220821,
0.023701859638094902,
0.01819240115582943,
-0.020863229408860207,
-0.006453338079154491,
-0.01455671712756157,
-0.028064679354429245,
0.023030657321214676,
-0.009173109196126461,
0.00983032863587141,
0.03135078027844429,
-0.007774769328534603,
0.006219116505235434,
-0.02019202709197998,
0.029532937332987785,
-0.028316380456089973,
-0.006530247163027525,
0.026834141463041306,
0.004579562693834305,
0.0014245586935430765,
-0.021688250824809074,
-2.9223117962828837e-05,
-0.016780078411102295,
-0.01075323298573494,
0.012627008371055126,
0.026134971529245377,
0.0004854861181229353,
-0.0015591488918289542,
0.003604220924898982,
-0.04259343072772026,
-0.03232961520552635,
0.007023161742836237,
0.023212440311908722,
0.018416134640574455,
-0.009466760791838169,
-0.01647244393825531,
0.021800117567181587,
0.016738127917051315,
0.0068938154727220535,
-0.00715250801295042,

-0.020989080891013145,
-0.02105899713933468,
0.0037929967511445284,
-0.008431988768279552,
0.016738127917051315,
0.19341836869716644,
0.020751362666487694,
0.02452688105404377,
0.03957301750779152,
-0.009858295321464539,
0.024904431775212288,
0.004862726666033268,
-0.004534116946160793,
0.008026470430195332,
0.03481866046786308,
0.02112891525030136,
-0.002181410090997815,
0.010914042592048645,
-0.009040267206728458,
-0.010921034030616283,
0.01912928931415081,
-0.04320869967341423,
-0.03459492698311806,
-0.029001567512750626,
-0.004873214289546013,
-0.007320309057831764,
-0.019940325990319252,
-0.019157255068421364,
-0.006554718129336834,
0.008802548982203007,
-0.005348649807274342,
-0.02363194338977337,
0.007935578934848309,
0.002069542882964015,
0.010809166356921196,
-0.007935578934848309,
0.005932456813752651,
0.00438379542902112,
-0.019884392619132996,
-0.014332982711493969,
-0.0045620836317539215,
0.02947700396180153,
0.007572010159492493,
0.029812604188919067,
0.0026830644346773624,
0.005086461082100868,
-0.03515426442027092,
0.021184848621487617,

-0.011522320099174976,
0.007159499917179346,
0.018569953739643097,
-0.01385754719376564,
-0.01117972657084465,
0.008872466161847115,
0.00025126419495791197,
-0.015997007489204407,
0.0014595171669498086,
0.0469282828271389,
0.01230539008975029,
0.030232107266783714,
0.007579001598060131,
0.005027031525969505,
-0.004597042221575975,
0.009550660848617554,
0.005247270222753286,
-0.02633073925971985,
-0.0004122917598579079,
-0.013948439620435238,
0.02550571784377098,
-0.04013235121965408,
-0.01782883331179619,
-0.025072233751416206,
-0.010935017839074135,
-0.018653852865099907,
-0.012864726595580578,
-0.006079282145947218,
-0.03853824362158775,
0.022471321746706963,
-0.009585619904100895,
0.007774769328534603,
-0.02669430710375309,
0.011536303907632828,
0.014458834193646908,
0.022177670150995255,
0.03557376563549042,
-0.004880206193774939,
-0.019157255068421364,
-0.005240278318524361,
-0.023687876760959625,
-0.006222612224519253,
-0.03244148567318916,
0.012165556661784649,
0.007222425192594528,
-0.017395347356796265,
-0.014277049340307713,
-0.030232107266783714,

-0.02533791773021221,
-0.0029959429521113634,
-0.006481305230408907,
0.010914042592048645,
0.02117086574435234,
-0.005904489662498236,
0.023687876760959625,
0.005740184802561998,
-0.004275423940271139,
-0.04186629503965378,
0.05000463128089905,
0.0035500351805239916,
-0.011186718940734863,
-0.012983584776520729,
0.0002704913495108485,
-0.005422062706202269,
-0.0039223432540893555,
-0.004359324462711811,
-0.025897253304719925,
0.004950122907757759,
-0.02093314751982689,
-0.004848743323236704,
-0.0015206945827230811,
0.010403648018836975,
0.012927651405334473,
0.012130597606301308,
-0.017031779512763023,
0.010354706086218357,
0.02178613469004631,
0.020989080891013145,
-0.00827817153185606,
0.01823435164988041,
0.009879270568490028,
0.00807541236281395,
0.0024470947682857513,
-0.016626261174678802,
-0.007606968749314547,
0.026764223352074623,
-0.024876466020941734,
-0.0034346722532063723,
-0.0035430435091257095,
-0.00017402776575181633,
-0.01978650875389576,
-0.0018877587281167507,
0.022093769162893295,
-0.004076160490512848,
-0.01322130300104618,
0.018010616302490234,

-0.012494166381657124,
0.013389104045927525,
0.008536865003407001,
-0.009732444770634174,
-0.020080160349607468,
-0.01279480941593647,
-0.010103004984557629,
0.016668209806084633,
0.012046697549521923,
0.005292716436088085,
-0.02694600820541382,
-0.03151857852935791,
-0.021450532600283623,
-0.010061055421829224,
-0.008942383341491222,
0.021898001432418823,
-0.024680698290467262,
-0.006932269781827927,
-0.04060778766870499,
-0.0034941015765070915,
0.016164807602763176,
-0.03839841112494469,
-0.002230352023616433,
0.020010242238640785,
-0.007453151047229767,
-0.05579375848174095,
-0.012025722302496433,
-0.17988243699073792,
0.007481118198484182,
0.0314067117869854,
-0.01549360528588295,
0.005761160049587488,
0.0037475507706403732,
0.01634659245610237,
-0.02276497147977352,
-0.007271367125213146,
-0.03470679372549057,
0.0425095297396183,
-0.01712966337800026,
-0.02358999289572239,
-0.01261302549391985,
-0.02537986822426319,
0.017800865694880486,
-0.008536865003407001,
0.026834141463041306,
0.034566961228847504,
0.010438607074320316,
0.028805799782276154,

-0.011983771808445454,
0.012172548100352287,
-0.013305203057825565,
0.004097135737538338,
-0.009152133949100971,
-0.00987227912992239,
0.0327770859003067,
0.010725266300141811,
-0.009466760791838169,
-0.0037195838522166014,
0.023072607815265656,
0.01638854295015335,
-0.013277236372232437,
0.03344829007983208,
-0.005722705740481615,
0.012969601899385452,
-0.011864913627505302,
0.006956740748137236,
0.011927838437259197,
-0.0026148955803364515,
0.027533311396837234,
0.005034023430198431,
-0.031378746032714844,
-0.019436923786997795,
0.021324682980775833,
0.020793313160538673,
-0.018178418278694153,
0.007107062265276909,
-0.04242562875151634,
0.009319934993982315,
-0.024093395099043846,
-0.011291594244539738,
-0.008858482353389263,
0.012263440527021885,
0.019283106550574303,
-0.007599976845085621,
0.013612838461995125,
0.010165930725634098,
-0.005292716436088085,
0.004978089593350887,
-0.021478500217199326,
0.0009508710936643183,
-0.004310382530093193,
0.009662528522312641,
-0.007935578934848309,
-0.014179165475070477,
0.017353396862745285,
-0.012165556661784649,

0.0038559220265597105,
-0.02714177593588829,
-0.01687796227633953,
0.033867791295051575,
-0.003196954494342208,
0.03045584075152874,
0.009298959746956825,
-0.009620578028261662,
0.019269121810793877,
0.020611528307199478,
0.008676698431372643,
-0.01619277521967888,
0.02841426432132721,
0.0059988778084516525,
0.003478370374068618,
0.016696177423000336,
0.004796305671334267,
-0.016849994659423828,
-0.003985268529504538,
0.0010767217027023435,
0.017143646255135536,
0.03305675461888313,
-0.008935391902923584,
-0.00852288119494915,
-0.009683502838015556,
0.010928025469183922,
0.0060478197410702705,
0.0024680697824805975,
0.013787630945444107,
0.0014359202468767762,
-0.006414883770048618,
0.005726201459765434,
0.011165743693709373,
-0.021450532600283623,
0.013612838461995125,
0.03613309934735298,
-0.00562132615596056,
-0.025156132876873016,
0.03232961520552635,
0.0219119843095541,
-0.019143272191286087,
-0.02607903815805912,
0.02187003567814827,
0.008334104903042316,
0.015465638600289822,
0.012668958865106106,
0.04063575342297554,
0.010837133973836899,

-0.03459492698311806,
-0.013521946035325527,
-0.0014804922975599766,
0.03663650155067444,
-0.01933903992176056,
-0.02297472395002842,
-0.002125476486980915,
-0.000681690638884902,
-0.013270244933664799,
-0.060072679072618484,
-0.030763475224375725,
0.012780825607478619,
0.011976780369877815,
-0.01806654967367649,
0.0003491479728836566,
-0.0006012861267663538,
0.03904164955019951,
-0.0026271308306604624,
-0.021226799115538597,
0.003716088132932782,
0.00987227912992239,
-0.002420875709503889,
0.00011809416901087388,
0.018164433538913727,
-0.0037929967511445284,
-0.023841693997383118,
9.750142635311931e-05,
-0.03168638050556183,
0.022918790578842163,
0.013892506249248981,
-0.02157638408243656,
0.012249456718564034,
-0.009795370511710644,
0.0007699608686380088,
0.015675390139222145,
-0.029672771692276,
0.018569953739643097,
0.015339788049459457,
-0.026400655508041382,
0.020611528307199478,
-0.02023397758603096,
1.3331340369404643e-06,
-0.014822402037680149,
-0.0010041827335953712,
0.009928212501108646,
-0.005348649807274342,
-0.011613212525844574,
0.03521019592881203,

-0.008222238160669804,
0.009243026375770569,
0.040943391621112823,
0.01708771288394928,
-0.04922156035900116,
-0.014332982711493969,
-0.016374560073018074,
0.0018440606072545052,
0.00548848370090127,
0.005894002504646778,
-0.015018169768154621,
-0.033252522349357605,
0.0032826026435941458,
-0.017996633425354958,
0.010914042592048645,
-0.0069672283716499805,
-0.00983032863587141,
0.013864539563655853,
0.023198457434773445,
0.00010968228161800653,
-0.037671275436878204,
-0.01117972657084465,
-0.012871718034148216,
-0.004348836839199066,
0.05020039901137352,
0.005048006772994995,
-0.0021551912650465965,
-0.0073622590862214565,
-0.021408582106232643,
0.011004934087395668,
-0.01522792037576437,
-0.005425558425486088,
-0.0076419273391366005,
-0.0017129662446677685,
-0.001426306669600308,
-0.025897253304719925,
0.010068046860396862,
-0.012724892236292362,
-0.006523255258798599,
0.007383234333246946,
-0.01140346098691225,
-0.024107377976179123,
-0.010005121119320393,
-0.01884962059557438,
-0.0018353209597989917,
0.018164433538913727,
0.024708664044737816,
0.006586180534213781,

0.021884018555283546,
0.03081941045820713,
-0.029504969716072083,
0.018611902371048927,
-0.005607342813163996,
0.03624496981501579,
-0.03755940869450569,
-0.013004560023546219,
0.01434696651995182,
0.027197709307074547,
-0.012515141628682613,
0.000878332182765007,
0.0023387232795357704,
-0.06689658015966415,
-0.0385102778673172,
-0.0840681865811348,
0.025295967236161232,
0.010438607074320316,
-0.0465087816119194,
-0.014892319217324257,
-0.029085468500852585,
-0.0018195896409451962,
-0.03344829007983208,
-0.011102817952632904,
0.022023852914571762,
-0.02174418419599533,
0.010886075906455517,
0.012039706110954285,
-0.012417257763445377,
-0.009131158702075481,
-0.026624390855431557,
0.02484849840402603,
-0.004775330424308777,
0.017367379739880562,
0.005338162183761597,
0.004785818047821522,
-0.0256735198199749,
0.016416508704423904,
0.02526800148189068,
-0.0035360518377274275,
0.03196604922413826,
-0.01455671712756157,
0.02575741894543171,
-0.0005807480192743242,
-0.014752484858036041,
0.01631862483918667,
-0.022918790578842163,
-0.02261115424335003,

0.010781199671328068,
0.018052566796541214,
0.004736876115202904,
-0.004750859457999468,
0.02199588529765606,
0.015913106501102448,
0.003478370374068618,
-0.027155758813023567,
-0.015633439645171165,
0.01252912450581789,
-0.0323016494512558,
-0.015507588163018227,
0.011046884581446648,
-0.0012384046567603946,
-0.0014079534448683262,
0.01117972657084465,
-0.012179539538919926,
0.025128167122602463,
0.015759289264678955,
-0.03445509448647499,
-0.03314065560698509,
0.004995569121092558,
-0.023044640198349953,
-0.002940009580925107,
0.016710160300135612,
-0.0030536246486008167,
0.004866222385317087,
0.03937724977731705,
0.004565579351037741,
0.01942294090986252,
0.0018388169119134545,
0.016654226928949356,
-0.005575879942625761,
-0.006708535365760326,
0.015395721420645714,
0.013375120237469673,
-0.02265310473740101,
-0.013619829900562763,
-0.026764223352074623,
-0.0030728518031537533,
0.012689934112131596,
0.01234734058380127,
-0.0034171929582953453,
-0.003918847534805536,
-0.03364405781030655,
0.005240278318524361,
0.0416145920753479,
0.009718461893498898,

-0.005582871846854687,
-0.013193336315453053,
0.019143272191286087,
0.026218872517347336,
0.005498971324414015,
-0.004114614799618721,
-0.008928399533033371,
-0.0013939699856564403,
-0.013542921282351017,
-0.03151857852935791,
0.02510019950568676,
0.004569075535982847,
0.020667461678385735,
-0.0011711096158251166,
0.017465263605117798,
-0.005159873981028795,
0.006201636977493763,
0.012431240640580654,
-0.010005121119320393,
-0.006215620320290327,
0.013787630945444107,
0.01946488954126835,
-0.03878994658589363,
-0.02031787671148777,
-0.005838068667799234,
-0.04885799437761307,
-0.003378738649189472,
-0.02080729603767395,
0.011697112582623959,
0.01647244393825531,
-0.010501531884074211,
0.004436233080923557,
0.005135403014719486,
-0.017605097964406013,
0.007236408535391092,
0.00023312946723308414,
-0.002006617607548833,
-0.03602123260498047,
0.025281984359025955,
0.00460752984508872,
0.001992634264752269,
0.04852239042520523,
-0.0019349526846781373,
0.001352893770672381,
0.004841751419007778,
0.0008564831223338842,
0.006806419230997562,
0.018304267898201942,

0.0038629136979579926,
-0.0003819215635303408,
-0.0010706039611250162,
-0.023701859638094902,
-0.027812978252768517,
-0.00788663700222969,
-0.011829954572021961,
-0.03442712500691414,
-0.01630464196205139,
0.006670081056654453,
0.08775980770587921,
0.012585058808326721,
-0.02010812610387802,
0.0219119843095541,
-0.032525382936000824,
0.02019202709197998,
-0.0005855548079125583,
0.010026096366345882,
0.0010950749274343252,
-0.022904805839061737,
-0.004579562693834305,
-0.015647422522306442,
0.005680755712091923,
-0.02930920198559761,
-0.027463393285870552,
0.007774769328534603,
-0.018695803359150887,
0.010235847905278206,
0.003401461523026228,
0.01778688281774521,
0.02427517995238304,
0.02412136271595955,
0.005467508919537067,
0.0018755232449620962,
-0.030595675110816956,
0.0037755174562335014,
0.016668209806084633,
-0.02276497147977352,
0.00539409602060914,
-0.002174418419599533,
0.007096574641764164,
0.0074601429514586926,
-0.02416331134736538,
-0.019646674394607544,
0.008306138217449188,
0.007369250990450382,
0.005404583178460598,
-0.014710535295307636,

```

        0.01900343783199787,
        -0.015913106501102448,
        0.02031787671148777,
        0.008089396171271801,
        -0.04374007135629654,
        -0.029197335243225098,
        -0.0012392787029966712,
        -0.030064305290579796,
        -0.0015888636698946357,
        -0.024415012449026108,
        -0.022750988602638245
    ]
}
},
{
    "_index": "github_issues",
    "_id": "BmLQUZMBKaR7I-arfsTE",
    "_score": 1.0,
    "_source": {
        "_type": "issue",
        "_repo": "langchain",
        "_issueNumber": "28272",
        "_title": "api-docs: add standard tests package to build",
        "_createdAt": "2024-11-21T23:33:36Z",
        "_closedAt": "2024-11-21T23:35:48Z",
        "_state": "closed",
        "_body": "None",
        "GitHub_Issue_vector": [
            -0.00266593461856246,
            -0.00042508705519139767,
            0.010304286144673824,
            -0.0032703070901334286,
            -0.028544705361127853,
            0.018860649317502975,
            -0.003763672197237611,
            -0.030560454353690147,
            -0.01621057279407978,
            -0.0016104852547869086,
            0.023033110424876213,
            -0.003592756576836109,
            0.00851407554000616,
            0.007858604192733765,
            -0.007273613940924406,
            -0.0017250165110453963,
            0.01566082425415516,
            -0.029432762414216995,
            0.014927824027836323,
            0.017761150375008583,

```

0.0171832088381052,
0.01380013208836317,
-0.00849997904151678,
0.011615228839218616,
-0.027952667325735092,
0.025175726041197777,
0.01018446870148182,
-0.025767764076590538,
0.007421623449772596,
0.00681196479126811,
0.03340787813067436,
-0.02600739896297455,
-0.006269263103604317,
-0.029066262766718864,
-0.003021862357854843,
-0.0018342616967856884,
0.018818361684679985,
0.00014327415556181222,
0.01386356446892023,
-0.012348228134214878,
0.011876007542014122,
0.00811938289552927,
0.0162669587880373,
-0.017042245715856552,
-0.014899631962180138,
-0.004221796989440918,
0.024499110877513885,
-0.012355276383459568,
-0.0009232978918589652,
0.01570311188697815,
0.017042245715856552,
-0.0045706769451498985,
0.003562802216038108,
0.020425323396921158,
0.017803438007831573,
0.011798478662967682,
0.0011461932444944978,
0.02060857228934765,
0.00804890226572752,
-0.020270265638828278,
-0.006808441132307053,
-0.008168719708919525,
-0.0271350909024477,
0.001348825404420495,
-0.01066373847424984,
-0.013764891773462296,
0.0034817492123693228,
0.011749141849577427,

0.012002872303128242,
-0.030616840347647667,
0.030193954706192017,
0.012580814771354198,
-0.0009144878131337464,
-0.013447728008031845,
0.04784233495593071,
-0.005959147587418556,
0.010297237895429134,
-0.022582033649086952,
-0.010614401660859585,
-0.009028584696352482,
0.025091148912906647,
0.0036121385637670755,
-0.019410399720072746,
0.02626112848520279,
0.0323929563164711,
0.00818986352533102,
0.013158757239580154,
0.019903765991330147,
0.00835196953266859,
-0.01699995808303356,
0.029601916670799255,
0.006223450414836407,
0.014448554255068302,
0.018804265186190605,
-0.00030416849767789245,
0.04600983485579491,
-0.012002872303128242,
0.050097718834877014,
0.03602976351976395,
-0.061797525733709335,
-0.001755851786583662,
0.009965978562831879,
-0.027882186695933342,
-0.00262717017903924,
-0.0005268436507321894,
-0.020030630752444267,
0.0006444583996199071,
0.0010880467016249895,
0.023061303421854973,
0.012073352932929993,
-0.028178205713629723,
0.039751145988702774,
0.0072524696588516235,
-0.05390368029475212,
0.023625148460268974,
-0.0035610401537269354,

-0.009331651963293552,
-0.03684733808040619,
-0.003421840723603964,
0.00547987874597311,
0.029489148408174515,
0.015618535690009594,
0.014434458687901497,
0.002662410493940115,
0.0003094545681960881,
0.004443811718374491,
-0.013743747025728226,
-0.029489148408174515,
-0.012757016345858574,
-0.009550142101943493,
0.015562150627374649,
0.009254123084247112,
-0.002803371986374259,
-0.012552622705698013,
-0.01608370803296566,
0.006008484400808811,
-0.008246248587965965,
0.020030630752444267,
-0.019664131104946136,
-0.02627522498369217,
-0.005737133324146271,
0.034479185938835144,
-0.02200409211218357,
-0.02128518745303154,
-0.030645031481981277,
0.02717737853527069,
0.01057916134595871,
-0.0005752991419285536,
0.022807572036981583,
-0.0335206463932991,
0.0025108768604695797,
-0.0223705917596817,
-0.007407527416944504,
-0.0009523712215013802,
-0.015068785287439823,
0.01604142040014267,
-0.011164151132106781,
0.012327084317803383,
-0.03321053087711334,
0.01026199758052826,
0.001115357968956232,
0.02104555256664753,
0.008774854242801666,
-0.03704468533396721,

0.005282532423734665,
0.029912032186985016,
0.018014881759881973,
-0.007093887776136398,
-0.004722210578620434,
-0.00559617206454277,
-0.01074126735329628,
0.0028157061897218227,
-0.04493853077292442,
0.024513207376003265,
-0.008521122857928276,
0.04454383626580238,
-0.0005616435082629323,
0.02197589911520481,
-0.018169937655329704,
0.006463085301220417,
-0.02595101296901703,
-0.0005590004730038345,
0.012665391899645329,
0.05711760371923447,
0.003774244338274002,
0.012320036068558693,
0.009839113801717758,
0.0025531654246151447,
0.023399610072374344,
0.006501849740743637,
0.024203091859817505,
0.020115207880735397,
-0.00828853715211153,
0.0009594192961230874,
-0.6094048023223877,
-0.010769459418952465,
-0.0024368721060454845,
-0.011291016824543476,
0.013891756534576416,
0.02765664830803871,
0.011868959292769432,
-0.002873852849006653,
0.0001169540046248585,
0.012270699255168438,
-0.02736062929034233,
0.02772712893784046,
0.019142573699355125,
-0.011403786018490791,
-0.0014272353146225214,
-0.009916641749441624,
0.034535568207502365,
-0.007470959797501564,

-0.029263610020279884,
0.013088276609778404,
-0.04600983485579491,
0.018494149670004845,
0.003326691687107086,
-0.005271960515528917,
0.02180674485862255,
-0.029404571279883385,
0.022342398762702942,
0.007703546434640884,
0.008831238374114037,
-0.0005616435082629323,
-0.023822495713829994,
0.023653341457247734,
0.00673443591222167,
-0.00042861109250225127,
0.045445989817380905,
0.004461431875824928,
-0.009169545955955982,
0.010825843550264835,
0.009190690703690052,
0.01578768901526928,
-0.02737472578883171,
0.006184686440974474,
-0.0008845334523357451,
0.00819691177457571,
-0.011143007315695286,
-0.010001218877732754,
0.020270265638828278,
-0.000212213140912354,
-0.006938830483704805,
0.009458517655730247,
0.035691455006599426,
-0.01639382354915142,
-0.004080835729837418,
0.011650469154119492,
-0.0019311726791784167,
-0.034620147198438644,
0.025133436545729637,
-0.029179032891988754,
0.006100109312683344,
0.008690277114510536,
0.007322950288653374,
0.005687796976417303,
0.0069775949232280254,
-0.020383033901453018,
-0.005560931749641895,
0.00841540191322565,

0.007365238852798939,
0.0260778795927763,
0.014427410438656807,
-0.045756105333566666,
-0.007358190603554249,
-0.017577899619936943,
-0.004669350106269121,
-0.005701893009245396,
0.014688189141452312,
0.028953494504094124,
0.018522342666983604,
0.011650469154119492,
-0.03608614578843117,
0.01582997664809227,
0.0010501632932573557,
-0.014300544746220112,
-0.019946053624153137,
-0.014124343171715736,
0.02597920596599579,
0.008739613927900791,
-0.01604142040014267,
0.00397863844409585,
0.003774244338274002,
-0.0111148152500391,
0.006255167070776224,
0.017056342214345932,
0.01582997664809227,
-0.005927431397140026,
0.0007845389191061258,
0.023047206923365593,
0.014194823801517487,
0.008485882543027401,
0.020101111382246017,
-0.025175726041197777,
-0.009916641749441624,
0.023470090702176094,
-0.0053917779587209225,
0.009282315149903297,
0.029122648760676384,
-0.009056776762008667,
-0.02754387818276882,
0.014025670476257801,
0.011720949783921242,
0.0025619755033403635,
-0.017507418990135193,
0.004746878985315561,
-0.002865042770281434,
0.018931129947304726,

-0.0034095065202564,
-0.04383902996778488,
-0.017324170097708702,
-0.02590872533619404,
0.004098455887287855,
0.018057169392704964,
-0.020256169140338898,
0.003562802216038108,
0.0010915707098320127,
0.02228601463139057,
0.017070438712835312,
0.001057211309671402,
-0.008655036799609661,
-0.017267784103751183,
-0.009923689998686314,
0.003155775833874941,
-0.005113378632813692,
0.02158120647072792,
0.018804265186190605,
-0.011072526685893536,
0.012651295401155949,
0.02239878475666046,
0.01659116894006729,
-0.009409180842339993,
0.015590342693030834,
-0.0002951381611637771,
-0.006649859249591827,
-0.003011290216818452,
0.004390951246023178,
-0.022412879392504692,
0.014300544746220112,
-0.040343184024095535,
-0.029714686796069145,
0.0005277246236801147,
-0.004711638670414686,
0.005962671712040901,
-0.017084535211324692,
-0.0070163593627512455,
-0.01691538095474243,
-0.01592865027487278,
-0.004271133802831173,
0.004426191560924053,
0.027698935940861702,
-0.03848249092698097,
-0.008091190829873085,
-2.0166857211734168e-05,
0.023737918585538864,
-0.009317555464804173,

-0.019283534958958626,
-0.012545574456453323,
-0.01649249717593193,
-0.008070046082139015,
-0.01585816964507103,
0.02072134241461754,
-0.006544138304889202,
-0.03343607112765312,
0.0014950729673728347,
-0.01633743941783905,
-0.006815488915890455,
-0.012094497680664062,
0.0037601483054459095,
0.002216619672253728,
-0.013356102630496025,
0.0032861651852726936,
-0.023103591054677963,
-0.005856950767338276,
-0.004278181586414576,
0.0016474876319989562,
-0.02637389861047268,
0.01294026616960764,
0.03228018432855606,
0.015125170350074768,
0.025654993951320648,
0.011150055564939976,
0.012369372881948948,
-0.005775897763669491,
-0.024273572489619255,
0.00670976797118783,
-0.012404613196849823,
0.0007946704863570631,
-0.0213133804500103,
0.002986622042953968,
-0.0011268110247328877,
0.005085186567157507,
0.00826034415513277,
0.023667437955737114,
-0.004546008538454771,
0.0069775949232280254,
-0.002498542657122016,
-0.015519862063229084,
0.007569632958620787,
-0.014645900577306747,
0.011150055564939976,
-0.02744520641863346,
0.009070873260498047,
-0.011805525980889797,

0.012771112844347954,
-0.017352361232042313,
-0.023512380197644234,
-0.00788679625838995,
0.0005977649125270545,
-0.006054296623915434,
-0.007344094570726156,
-0.020383033901453018,
-0.01694357395172119,
0.0015082882018759847,
-0.013433631509542465,
-0.003883489640429616,
0.0038729174993932247,
-0.0019346966873854399,
0.013856516219675541,
-0.014392170123755932,
0.010677834041416645,
0.03834152966737747,
0.02122880332171917,
0.008711420930922031,
-0.002364629413932562,
0.023596957325935364,
0.005606743972748518,
-0.007322950288653374,
0.023089494556188583,
0.015153362415730953,
0.024597782641649246,
-0.017408747225999832,
0.024062128737568855,
-0.025344880297780037,
0.019692322239279747,
0.017169112339615822,
0.01677441969513893,
-0.01325742993503809,
0.03664999082684517,
0.0006070155068300664,
0.025429455563426018,
0.00836606603115797,
0.011713901534676552,
-0.007287709973752499,
0.006914162077009678,
-0.011516555212438107,
-0.00020923973352182657,
-0.009077921509742737,
0.014984208159148693,
-0.017648380249738693,
-0.029658300802111626,
0.034845683723688126,

-0.0015902220038697124,
0.02128518745303154,
0.019128477200865746,
-0.014067959040403366,
0.025387167930603027,
0.02111603319644928,
0.03354883939027786,
-0.014237112365663052,
-0.0078163156285882,
-0.017620189115405083,
0.01611190102994442,
-0.0072066569700837135,
-7.246302266139537e-05,
-0.007421623449772596,
0.004013878758996725,
-0.019889669492840767,
0.05263502895832062,
0.011720949783921242,
0.0042922780849039555,
0.02626112848520279,
0.0271350909024477,
-0.028121821582317352,
-0.030081186443567276,
-0.024329956620931625,
-0.0003213481977581978,
0.008302632719278336,
-0.004013878758996725,
-0.004503720439970493,
-0.024682359769940376,
0.0008299108594655991,
-0.025387167930603027,
0.016901284456253052,
-0.0015549816889688373,
-0.008584556169807911,
-0.011361497454345226,
-0.022737091407179832,
-0.015139265917241573,
-0.0019311726791784167,
0.023145880550146103,
-0.01643611118197441,
0.007315902505069971,
0.02083411067724228,
0.0020738961175084114,
0.003546943888068199,
0.029742877930402756,
-0.007428671699017286,
0.022863956168293953,
0.015280227176845074,

0.00208270619623363,
-0.03546591475605965,
0.0019117904594168067,
-0.002216619672253728,
0.02125699631869793,
0.007851555943489075,
0.011756190098822117,
-0.007485056295990944,
-0.008683228865265846,
0.0031029151286929846,
-0.0028350884094834328,
-0.014279400929808617,
0.017535611987113953,
0.0017593757947906852,
-0.0165488813072443,
-0.0330977626144886,
-0.003206874243915081,
0.0039997827261686325,
0.023822495713829994,
0.04772956669330597,
0.011361497454345226,
0.004454383626580238,
-0.01598503440618515,
0.02096097730100155,
-0.011622276157140732,
0.009958930313587189,
-0.001368207624182105,
-0.018747881054878235,
-0.019621841609477997,
-0.006797868758440018,
-0.00405616732314229,
0.006212878506630659,
0.03202645480632782,
0.02055218815803528,
-0.012186122126877308,
-0.017126822844147682,
-0.025316687300801277,
-0.0007034860318526626,
0.0006669241702184081,
-0.03244933858513832,
0.004686970263719559,
0.006487753242254257,
0.045192260295152664,
-0.006026104558259249,
-0.025232110172510147,
0.013743747025728226,
0.025584513321518898,
-0.012869785539805889,

0.010290190577507019,
0.0026676966808736324,
0.0006629595882259309,
0.030560454353690147,
-0.02643028274178505,
0.040963415056467056,
-0.003044768702238798,
0.00034711771877482533,
-0.003060626797378063,
0.02065086178481579,
0.0132503816857934,
-0.001957603031769395,
-0.00547987874597311,
-0.02108784206211567,
0.017591996118426323,
0.005892191082239151,
-0.01637972705066204,
0.00804890226572752,
-0.0055855996906757355,
-0.019128477200865746,
0.02717737853527069,
0.020227976143360138,
-0.03915910795331001,
-0.03149080276489258,
0.03729841485619545,
0.025697283446788788,
-0.025767764076590538,
0.019410399720072746,
-0.02074953354895115,
-0.013666218146681786,
-0.03205464780330658,
0.009691103361546993,
-0.025232110172510147,
-0.0206790529191494,
-0.030785992741584778,
-0.004827931523323059,
-0.02180674485862255,
-0.011206439696252346,
-0.03735480085015297,
0.009148402139544487,
0.009832065552473068,
-0.013292670249938965,
-0.005028801970183849,
0.014053862541913986,
0.022722994908690453,
-0.0007215467048808932,
0.01584407314658165,
-0.011495411396026611,

0.009606527164578438,
0.02686726301908493,
-0.022962629795074463,
-0.02170807309448719,
0.011805525980889797,
0.0007554655894637108,
0.02146843820810318,
0.012277747504413128,
-0.0012351751793175936,
-0.004161888733506203,
-0.0026254081167280674,
0.03580422326922417,
-0.02156711183488369,
0.017338264733552933,
-0.0056278882548213005,
-0.035296760499477386,
0.012834545224905014,
0.00210208841599524,
0.003714335849508643,
0.017648380249738693,
-0.018959322944283485,
0.011643420904874802,
-0.0066392868757247925,
-0.03247753158211708,
-0.019635938107967377,
-0.024414533749222755,
-0.006551186088472605,
-0.0018272135639563203,
-0.012242507189512253,
0.0159145537763834,
-0.033971723169088364,
0.017239592969417572,
0.003543419996276498,
-0.018902938812971115,
-0.007400479167699814,
0.0069670225493609905,
0.009099065326154232,
0.019762802869081497,
0.011460171081125736,
0.02149662934243679,
-0.010445247404277325,
-0.008866478689014912,
0.012489189393818378,
-0.03749576210975647,
0.018846554681658745,
0.03146260976791382,
0.0061106812208890915,
0.020044727250933647,

-0.01071307435631752,
-0.008767805993556976,
-0.02662762999534607,
-0.005398825742304325,
0.009846161119639874,
0.015252035111188889,
0.008965152315795422,
-0.006847205106168985,
-0.0005907168379053473,
-0.03290041536092758,
0.006107157561928034,
0.0010096367914229631,
-0.00533186923712492,
-0.004450859501957893,
-0.015181554481387138,
0.0031892540864646435,
-0.018508246168494225,
-0.02165168710052967,
0.0017858061473816633,
-0.03882079944014549,
0.0016219383105635643,
0.009282315149903297,
-0.0007352023967541754,
0.02136976458132267,
-0.029855648055672646,
-0.004997085314244032,
0.010219709016382694,
0.012038112618029118,
-0.015576247125864029,
-0.019988341256976128,
0.008619796484708786,
-0.007104460150003433,
0.034450992941856384,
0.03304137662053108,
0.023667437955737114,
-0.00789384450763464,
0.020185688510537148,
-0.011495411396026611,
0.029150839895009995,
-0.013926996849477291,
-0.0266417246311903,
-0.00021815995569340885,
-0.00656175846233964,
-0.00821100827306509,
0.017451034858822823,
0.01564672775566578,
0.0029883841052651405,
0.0054270182736217976,

-0.018860649317502975,
0.025711379945278168,
-0.0223705917596817,
-0.0017003482207655907,
-0.022708898410201073,
-0.04491033777594566,
-0.022948533296585083,
0.009077921509742737,
-0.040286798030138016,
0.024400437250733376,
-0.045389607548713684,
-0.019339919090270996,
0.05813252553343773,
-0.002117946743965149,
0.0028562326915562153,
-0.009211834520101547,
0.0318291075527668,
-0.023681534454226494,
0.0156326312571764,
0.0037953886203467846,
0.006385556422173977,
-0.020030630752444267,
0.01324333343654871,
-0.008605699986219406,
-0.006995215080678463,
0.008929912000894547,
0.014223015867173672,
0.019635938107967377,
-2.0318280803621747e-05,
0.013236286118626595,
-0.024724649265408516,
0.0426267609000206,
-0.02724785916507244,
-0.0036473791114985943,
-0.02121470682322979,
-0.023568764328956604,
-0.0022430499084293842,
-0.019269438460469246,
0.01356754545122385,
-0.022497456520795822,
0.02110193856060505,
0.010177421383559704,
-4.3307121813995764e-05,
0.01590045914053917,
-0.03292860835790634,
0.007513248361647129,
0.019748708233237267,
-0.009500805288553238,

0.04195014387369156,
0.034084491431713104,
0.018860649317502975,
0.024033937603235245,
0.017747053876519203,
-0.02624703384935856,
-0.0037883406039327383,
-0.02603559009730816,
-0.008464738726615906,
0.02158120647072792,
0.016985861584544182,
-0.009324603714048862,
-0.031152494251728058,
0.019819188863039017,
0.011749141849577427,
-0.008781902492046356,
-0.040850646793842316,
0.005261388141661882,
0.003265020903199911,
0.012545574456453323,
-0.02726195566356182,
0.012136786244809628,
-0.025499936193227768,
0.012693583965301514,
-0.009712248109281063,
-0.007245421409606934,
-0.011312161572277546,
-0.01320104580372572,
0.012792256660759449,
0.015139265917241573,
-0.007139700464904308,
0.03777768462896347,
0.023554667830467224,
-0.030109377577900887,
0.007668306119740009,
-0.0017919731326401234,
-0.0016809660010039806,
0.019833283498883247,
0.019057996571063995,
0.039441030472517014,
0.0020809443667531013,
0.0020210356451570988,
0.01666164956986904,
0.005194431636482477,
-0.007421623449772596,
-0.002976049901917577,
0.019184861332178116,
0.040343184024095535,

-0.02073543891310692,
0.006121253594756126,
0.008598651736974716,
-0.01354640070348978,
0.002177855232730508,
-0.018141746520996094,
-0.0128627372905612,
-0.020115207880735397,
-0.012031064368784428,
-0.0033090715296566486,
0.012059257365763187,
0.019198957830667496,
-0.002117946743965149,
-0.013208093121647835,
-0.013588689267635345,
-0.004260561428964138,
-0.0265007633715868,
-0.014504939317703247,
0.007703546434640884,
-0.02200409211218357,
-0.034817494451999664,
-0.004412095062434673,
-0.02719147503376007,
0.014547227881848812,
0.0017549707554280758,
0.008929912000894547,
-0.015407092869281769,
0.040540531277656555,
-0.000717582181096077,
0.029263610020279884,
-0.020016534253954887,
-0.02630341798067093,
-0.010191516950726509,
0.00828853715211153,
-0.012707680463790894,
-0.019283534958958626,
0.0072524696588516235,
-0.015082881785929203,
-0.04288049042224884,
-0.005219100043177605,
0.01072012260556221,
0.023625148460268974,
-0.007414575200527906,
-0.028446031734347343,
0.007442767731845379,
0.01118529587984085,
0.009225931018590927,
-0.006068393122404814,

-0.024513207376003265,
0.04908279702067375,
-0.019903765991330147,
-0.0004546008713077754,
-0.00032090768218040466,
-0.008598651736974716,
-0.025288494303822517,
0.0037460520397871733,
-6.502951146103442e-05,
0.007449815981090069,
-0.01656297780573368,
-0.029996609315276146,
0.0033425497822463512,
0.0014219492441043258,
0.0008673538104631007,
-0.029545532539486885,
-0.014575419947504997,
-0.039779338985681534,
-0.025415360927581787,
0.028727956116199493,
0.01578768901526928,
0.003562802216038108,
0.001784925116226077,
0.007992517203092575,
0.03306956961750984,
0.034479185938835144,
-0.028615185990929604,
-0.006614618934690952,
-0.028953494504094124,
-0.008605699986219406,
-0.014716381207108498,
-0.029968416318297386,
-0.018494149670004845,
0.023216361179947853,
0.035324953496456146,
0.009789776988327503,
-0.0111148152500391,
-0.012749968096613884,
-0.007196085061877966,
0.0006043724715709686,
-0.008154623210430145,
-0.012700632214546204,
0.010064652189612389,
-0.00788679625838995,
-0.01348296832293272,
0.017225496470928192,
-0.006191734224557877,
0.019692322239279747,

0.0008316729217767715,
-0.023075399920344353,
-0.00033588483347557485,
0.00018247908155899495,
0.014131391420960426,
-0.0322519950568676,
-0.0046164896339178085,
0.005564455408602953,
0.020087014883756638,
0.018959322944283485,
0.02239878475666046,
0.013743747025728226,
-0.00405616732314229,
-0.008464738726615906,
-0.029517339542508125,
-0.006847205106168985,
-0.015040593221783638,
0.01608370803296566,
0.006075440905988216,
-0.01687309332191944,
-0.019819188863039017,
0.01629514992237091,
0.006945878267288208,
0.02774122543632984,
-0.0530015267431736,
0.006653383374214172,
0.023653341457247734,
0.019368112087249756,
-0.023033110424876213,
-0.013158757239580154,
-0.014547227881848812,
-0.030673224478960037,
0.024527302011847496,
-0.010769459418952465,
0.01036771945655346,
0.03298499435186386,
-0.012510334141552448,
0.003936350345611572,
-0.022328302264213562,
0.024964284151792526,
-0.028925301507115364,
-0.005257864482700825,
0.02136976458132267,
0.0007281543221324682,
-0.0015267893904820085,
-0.02194770611822605,
7.361935422522947e-05,
-0.009430324658751488,

-0.007936133071780205,
0.010614401660859585,
0.030785992741584778,
0.0017153254011645913,
-0.0013990430161356926,
0.00812643114477396,
-0.03743937611579895,
-0.02626112848520279,
0.013081228360533714,
0.022525649517774582,
0.018085362389683723,
-0.011699805036187172,
-0.014342833310365677,
0.01701405458152294,
0.017422841861844063,
0.008464738726615906,
-0.014843246899545193,
-0.020326649770140648,
-0.02210276387631893,
0.0045319125056266785,
-0.004894888494163752,
0.012108593247830868,
0.191820427775383,
0.02210276387631893,
0.025866437703371048,
0.03732660785317421,
-0.012813401408493519,
0.025514032691717148,
1.92996121768374e-05,
-0.010247902013361454,
0.004419143311679363,
0.03304137662053108,
0.02146843820810318,
-0.0009188928524963558,
0.01063554547727108,
-0.008711420930922031,
-0.012982554733753204,
0.018945226445794106,
-0.044825758785009384,
-0.03580422326922417,
-0.02767074480652809,
-0.008936959318816662,
-0.007611921522766352,
-0.02062266878783703,
-0.017817534506320953,
-0.002819230081513524,
0.0027998480945825577,
-0.005560931749641895,

-0.019509073346853256,
0.011608180589973927,
0.0018148794770240784,
0.012002872303128242,
-0.007287709973752499,
0.01056506484746933,
0.007534392643719912,
-0.023145880550146103,
-0.009874354116618633,
-0.008549315854907036,
0.025429455563426018,
0.007618969772011042,
0.02729014866054058,
0.006177638191729784,
0.002322340849786997,
-0.03594518452882767,
0.028713859617710114,
-0.01364507433027029,
0.0031222973484545946,
0.023907072842121124,
-0.010522776283323765,
-0.012686535716056824,
0.01112186349928379,
-0.0008351969299837947,
-0.014603612013161182,
-0.0016430825926363468,
0.05288875848054886,
0.013856516219675541,
0.028262782841920853,
0.001650130725465715,
0.0030800090171396732,
-0.0040033068507909775,
0.011868959292769432,
0.009958930313587189,
-0.024188995361328125,
0.002517924876883626,
-0.013313814997673035,
0.028671570122241974,
-0.039976682513952255,
-0.018310900777578354,
-0.02665582112967968,
-0.007062171585857868,
-0.012453949078917503,
-0.01080469973385334,
-0.014124343171715736,
-0.03811599314212799,
0.019706418737769127,
-0.012369372881948948,

0.006551186088472605,
-0.029038071632385254,
0.01578768901526928,
0.017535611987113953,
0.023484187200665474,
0.03927187621593475,
0.000467816018499434,
-0.023752015084028244,
-0.007590777240693569,
-0.018339091911911964,
-0.008168719708919525,
-0.028431937098503113,
0.010036459192633629,
0.006914162077009678,
-0.020397130399942398,
-0.01064259372651577,
-0.034845683723688126,
-0.02638799510896206,
-0.0023981076665222645,
-0.008020710200071335,
0.008894671685993671,
0.024442724883556366,
-0.0021003265865147114,
0.0166898425668478,
0.004472003784030676,
0.00020615621178876609,
-0.040596913546323776,
0.04358530044555664,
-0.002325864974409342,
-0.012200218625366688,
-0.009056776762008667,
0.0018606919329613447,
-0.00346765317954123,
-0.003989210817962885,
-0.0076964981853961945,
-0.03261849284172058,
0.006766152568161488,
-0.02198999561369419,
-0.0002872090844903141,
-0.004912508651614189,
0.015604439191520214,
0.013208093121647835,
0.01319399755448103,
-0.017676573246717453,
0.009705199860036373,
0.024682359769940376,
0.020185688510537148,
-0.009183642454445362,

0.020397130399942398,
0.004757450893521309,
0.011157103814184666,
0.005723037291318178,
-0.017465131357312202,
-0.008218055590987206,
0.028290975838899612,
-0.024569591507315636,
-0.0025514033623039722,
-0.007922036573290825,
-0.005229671951383352,
-0.024259475991129875,
-0.00826034415513277,
0.024118514731526375,
-0.003559278091415763,
-0.014786862768232822,
0.019156668335199356,
-0.01372260320931673,
0.013849467970430851,
0.008683228865265846,
-0.012305939570069313,
-0.02182084135711193,
-0.017267784103751183,
-0.005772373639047146,
0.019100284203886986,
0.012644247151911259,
0.004323994275182486,
-0.022680707275867462,
-0.0325339175760746,
-0.025499936193227768,
-0.008901719003915787,
-0.009846161119639874,
0.02626112848520279,
-0.02707870677113533,
-0.00836606603115797,
-0.040174029767513275,
-0.004059691447764635,
0.020566284656524658,
-0.03786225989460945,
-0.004109027795493603,
0.023145880550146103,
-0.011453122831881046,
-0.055567026138305664,
-0.009120210073888302,
-0.18155843019485474,
0.010628498159348965,
0.030081186443567276,
-0.017563804984092712,

0.0077881235629320145,
0.0017100393306463957,
0.018339091911911964,
-0.020298456773161888,
-0.010170373134315014,
-0.03653722256422043,
0.04299325868487358,
-0.01588636264204979,
-0.029489148408174515,
-0.008781902492046356,
-0.02688135951757431,
0.011291016824543476,
-0.008401306346058846,
0.02721966803073883,
0.03205464780330658,
0.01602732390165329,
0.02599330246448517,
-0.012517382390797138,
0.007329998537898064,
-0.011227584443986416,
0.00414074445143342,
-0.009352795779705048,
-0.012298891320824623,
0.03132164850831032,
0.011495411396026611,
-0.009564238600432873,
-0.002250098157674074,
0.02066495642066002,
0.01064964197576046,
-0.018141746520996094,
0.03388714790344238,
-0.001252795453183353,
0.012876833789050579,
-0.012024017050862312,
0.006716815754771233,
0.012742920778691769,
-0.0042922780849039555,
0.024541398510336876,
0.0033636940643191338,
-0.03160357102751732,
-0.017535611987113953,
0.017436938360333443,
0.02155301533639431,
-0.018198130652308464,
0.006068393122404814,
-0.0437544509768486,
0.006945878267288208,
-0.028544705361127853,

-0.009000392630696297,
-0.0083449212834239,
0.012009920552372932,
0.02184903435409069,
-0.003085294971242547,
0.011495411396026611,
0.010191516950726509,
-0.008944007568061352,
0.008302632719278336,
-0.022835765033960342,
0.0020774202421307564,
-0.002322340849786997,
0.011128910817205906,
-0.004436763469129801,
-0.013299718499183655,
0.017620189115405083,
-0.012996651232242584,
0.0030236244201660156,
-0.022849861532449722,
-0.017775246873497963,
0.03137803077697754,
-0.0005871927714906633,
0.02734653279185295,
0.014053862541913986,
-0.007393431384116411,
0.017324170097708702,
0.02113012969493866,
0.011481314897537231,
-0.01618238165974617,
0.03191368654370308,
0.0020245597697794437,
0.0006620786152780056,
0.017958495765924454,
-0.00030394826899282634,
-0.01580178551375866,
-0.0066850995644927025,
-7.169214222813025e-05,
0.015082881785929203,
0.03385895490646362,
-0.005994388367980719,
-0.011636372655630112,
-0.005881619174033403,
0.012009920552372932,
0.008626844733953476,
-0.002197237452492118,
0.00686482572928071,
-0.001073950552381575,
-0.0063961283303797245,

0.0035081796813756227,
0.006589950527995825,
-0.018000785261392593,
0.015195650979876518,
0.034930262714624405,
-0.0045283883810043335,
-0.02610607072710991,
0.0326748788356781,
0.023455996066331863,
-0.02637389861047268,
-0.029460955411195755,
0.020340746268630028,
0.005571503657847643,
0.018564630299806595,
0.009662911295890808,
0.03611433878540993,
0.011453122831881046,
-0.03208284080028534,
-0.014117294922471046,
-0.002169045154005289,
0.03681914508342743,
-0.018677400425076485,
-0.024879707023501396,
-0.0022536220494657755,
-0.002056275960057974,
-0.0168308038264513,
-0.05869637429714203,
-0.029658300802111626,
0.009994170628488064,
0.008781902492046356,
-0.01580178551375866,
0.00021441567514557391,
0.0010589733719825745,
0.04434649273753166,
0.0003444746835157275,
-0.027994954958558083,
0.0034940834157168865,
0.004556580912321806,
-0.00650889752432704,
-0.0015426474856212735,
0.009627670980989933,
-0.009380988776683807,
-0.024752840399742126,
-0.004560105036944151,
-0.03349245339632034,
0.022934436798095703,
0.017662476748228073,
-0.023047206923365593,

0.012390516698360443,
-0.01073421910405159,
-0.00024381936236750335,
0.01580178551375866,
-0.029545532539486885,
0.019043900072574615,
0.020383033901453018,
-0.024823321029543877,
0.02125699631869793,
-0.018959322944283485,
-0.0005396182532422245,
-0.014674093574285507,
-0.0008021590765565634,
0.009684056043624878,
-0.007358190603554249,
-0.006498325616121292,
0.03301318734884262,
-0.009973026812076569,
0.012207266874611378,
0.04144268482923508,
0.01680261269211769,
-0.04854714497923851,
-0.012975506484508514,
-0.019494976848363876,
-0.00393987400457263,
0.0072595179080963135,
0.004623537417501211,
-0.009303459897637367,
-0.02754387818276882,
0.003802436636760831,
-0.017888015136122704,
0.010846988297998905,
-0.011319208890199661,
-0.012186122126877308,
0.012841593474149704,
0.0206790529191494,
-8.755031558393966e-06,
-0.040033068507909775,
-0.011706853285431862,
-0.012193170376121998,
-0.002900283085182309,
0.055764373391866684,
0.006188210099935532,
-0.0005708941025659442,
-0.0038588212337344885,
-0.02142614871263504,
0.01118529587984085,
-0.017747053876519203,

-0.0008783664088696241,
-0.007675354368984699,
-0.0013021319173276424,
0.00023589027114212513,
-0.02600739896297455,
0.006234022788703442,
-0.011967631988227367,
-0.007266565691679716,
0.00783746037632227,
-0.00783746037632227,
-0.02589462883770466,
-0.008626844733953476,
-0.017549708485603333,
-0.001364683615975082,
0.012890930287539959,
0.030250340700149536,
0.009275266900658607,
0.0257254745811224,
0.028234589844942093,
-0.02738882042467594,
0.018240420147776604,
-0.0066780513152480125,
0.033802568912506104,
-0.03746756911277771,
-0.01614009216427803,
0.012390516698360443,
0.02659943699836731,
-0.011608180589973927,
0.0004726615734398365,
0.005610268097370863,
-0.06839452683925629,
-0.03419726341962814,
-0.08113744854927063,
0.02627522498369217,
0.00826739240437746,
-0.040314991027116776,
-0.017155015841126442,
-0.0328158400952816,
-0.0037284318823367357,
-0.03763672336935997,
-0.009951882995665073,
0.02163759246468544,
-0.01692947745323181,
0.005508070811629295,
0.015096977353096008,
-0.014589516445994377,
-0.01056506484746933,
-0.022835765033960342,

0.024710552766919136,
-0.006568806245923042,
0.0165488813072443,
0.008781902492046356,
0.00539530161768198,
-0.022342398762702942,
0.015139265917241573,
0.02105964906513691,
-0.002764607546851039,
0.024400437250733376,
-0.013067131862044334,
0.02760026417672634,
-0.0039610182866454124,
-0.017408747225999832,
0.011988776735961437,
-0.023864783346652985,
-0.025034764781594276,
0.01081174798309803,
0.017324170097708702,
0.011777333915233612,
-0.005592647939920425,
0.02148253470659256,
0.019875572994351387,
0.0007391669205389917,
-0.028826627880334854,
-0.008831238374114037,
0.010522776283323765,
-0.03321053087711334,
-0.013990430161356926,
0.012531477957963943,
-0.001127692055888474,
-0.0031152493320405483,
0.009881401434540749,
-0.014032718725502491,
0.023145880550146103,
0.018254514783620834,
-0.034281838685274124,
-0.03335149213671684,
0.007238373626023531,
-0.019974246621131897,
0.0015391234774142504,
0.019241245463490486,
-0.007097411900758743,
0.0061987824738025665,
0.03898995369672775,
0.006984642706811428,
0.014716381207108498,
0.0004691375361289829,

0.0159145537763834,
-0.006452512927353382,
-0.009352795779705048,
0.018719688057899475,
0.0128627372905612,
-0.02091868780553341,
-0.01673213206231594,
-0.028121821582317352,
-0.002399869728833437,
0.015054688788950443,
0.019015707075595856,
-0.004429715219885111,
-0.005416445899754763,
-0.03349245339632034,
0.00850702729076147,
0.04352891445159912,
0.007428671699017286,
-0.0027152709662914276,
-0.009965978562831879,
0.017888015136122704,
0.02654305286705494,
0.0034271266777068377,
-0.002798086032271385,
-0.014476747252047062,
-0.0009576572338119149,
-0.014504939317703247,
-0.035719648003578186,
0.020242072641849518,
0.005205003544688225,
0.025203917175531387,
-0.0029690018855035305,
0.017944399267435074,
-0.006244594696909189,
0.008253296837210655,
0.011720949783921242,
-0.00509223435074091,
-0.0032703070901334286,
0.01026199758052826,
0.020171592012047768,
-0.03580422326922417,
-0.02643028274178505,
-0.0012677726335823536,
-0.04366987571120262,
-0.0014633566606789827,
-0.022694803774356842,
0.006043724715709686,
0.017253689467906952,
-0.007407527416944504,

0.008817142806947231,
0.0024421580601483583,
-0.014187775552272797,
0.011854862794280052,
0.0038482490926980972,
0.0012536763679236174,
-0.035578686743974686,
0.02683907188475132,
0.00811938289552927,
0.004979465156793594,
0.0443182997405529,
-0.004447335842996836,
0.002017511520534754,
0.009359844028949738,
-0.0005752991419285536,
0.003524037776514888,
0.019368112087249756,
0.0037284318823367357,
0.0004305933543946594,
-0.004401523154228926,
-0.023737918585538864,
-0.024978378787636757,
-0.01326447818428278,
-0.012975506484508514,
-0.030757801607251167,
-0.0166898425668478,
0.00810528639703989,
0.0858173668384552,
0.005486926529556513,
-0.01619647815823555,
0.024865610525012016,
-0.02730424515902996,
0.019706418737769127,
2.6237561542075127e-05,
0.005303676705807447,
-0.0021267568226903677,
-0.02207457274198532,
-0.0027804658748209476,
-0.011629324406385422,
0.0030165764037519693,
-0.03839791566133499,
-0.029996609315276146,
0.00821100827306509,
-0.014046814292669296,
0.015477573499083519,
0.00199284334667027,
0.01629514992237091,
0.03304137662053108,

```

        0.02221553400158882,
        0.005941527429968119,
        0.004221796989440918,
        -0.030955146998167038,
        0.008760757744312286,
        0.014815054833889008,
        -0.020227976143360138,
        0.008471786975860596,
        -0.0013162280665710568,
        0.004718686453998089,
        0.010128084570169449,
        -0.01657707430422306,
        -0.017225496470928192,
        0.00803480576723814,
        0.006265738978981972,
        0.007872700691223145,
        -0.010205613449215889,
        0.01691538095474243,
        -0.018536439165472984,
        0.017704766243696213,
        0.00789384450763464,
        -0.040343184024095535,
        -0.028220493346452713,
        7.433517021127045e-05,
        -0.0322519950568676,
        0.0024721124209463596,
        -0.017239592969417572,
        -0.025767764076590538
    ]
}
},
{
  "_index": "github_issues",
  "_id": "B2LQUZMBKaR7I-arfsTE",
  "_score": 1.0,
  "_source": {
    "_type": "issue",
    "_repo": "langchain",
    "_issueNumber": "28271",
    "_title": "PydanticUndefinedAnnotation: name 'SafetySetting' is not defined using ChatVertexAI",
    "_createdAt": "2024-11-21T22:43:53Z",
    "_closedAt": "2024-12-31T00:36:30Z",
    "_state": "open",
    "_body": "### Checked other resources\n\n- [X] I added a very descriptive title to this issue.\n- [X] I searched the LangChain documentation with the integrated search.\n- [X] I used the GitHub search to find a similar question and didn't find it.\n- [X] I am sure that this is a bug in LangChain"
  }
}

```

```
rather than my code.\n- [X] The bug is not resolved by updating to the latest\nstable version of LangChain (or the specific integration package).\n\n###\nExample Code\n\n```python\n# Ensure your VertexAI credentials are\nconfigured\nfrom langchain_google_vertexai import\nChatVertexAI\nmodel =\nChatVertexAI(model=\"gemini-1.5-flash\")\nmodel.invoke(\"Hello,\nworld!\")\n``` Error Message and Stack Trace (if applicable)\n\npython\n-----\n----\r\nNameError\nTraceback (most recent call\nlast)\r\nFile /opt/conda/lib/python3.10/site-\npackages/pydantic/_internal/_generate_schema.py:815, in\nGenerateSchema._resolve_forward_ref(self, obj)\r\n      814     try:\r\n--> 815         obj = _typing_extra.eval_type_backport(obj, *self._types_namespace)\r\n      816 except NameError as e:\r\nFile /opt/conda/lib/python3.10/site-\npackages/pydantic/_internal/_typing_extra.py:534, in eval_type_backport(value,\nglobals, locals, type_params)\r\n      533     try:\r\n--> 534         return\n_eval_type_backport(value, globals, locals, type_params)\r\n      535 except\nTypeError as e:\r\nFile /opt/conda/lib/python3.10/site-\npackages/pydantic/_internal/_typing_extra.py:558, in _eval_type_backport(value,\nglobals, locals, type_params)\r\n      557     try:\r\n--> 558         return\n_eval_type(value, globals, locals, type_params)\r\n      559 except TypeError as\ne:\r\nFile /opt/conda/lib/python3.10/site-\npackages/pydantic/_internal/_typing_extra.py:592, in _eval_type(value, globals,\nlocals, type_params)\r\n      591     else:\r\n--> 592         return typing._eval_type(\n# type: ignore\r\n      593             value, globals, locals)\r\n      594\r\n)\r\nFile /opt/conda/lib/python3.10/typing.py:327, in _eval_type(t,\nglobals, locals, recursive_guard)\r\n      326     if isinstance(t,\nForwardRef):\r\n--> 327         return t._evaluate(globals, locals,\nrecursive_guard)\r\n      328     if isinstance(t, (_GenericAlias, GenericAlias,\ntypes.UnionType)):\r\nFile /opt/conda/lib/python3.10/typing.py:699, in\nForwardRef._evaluate(self, globals, locals, recursive_guard)\r\n      693     type_ = _type_check(\r\n      694         eval(self.__forward_code__, globals,\nlocals),\r\n      695         \"Forward references must evaluate to types.\",\r\n      696         is_argument=self.__forward_is_argument__,\r\n      697         allow_special_forms=self.__forward_is_class__,\r\n      698     )\r\n--> 699\nself.__forward_value__ = _eval_type(\r\n      700         type_, globals, locals,\nrecursive_guard | {self.__forward_arg__},\r\n      701     )\r\n      702\nself.__forward_evaluated__ = True\r\nFile /opt/conda/lib/python3.10/typing.py:329, in _eval_type(t, globals, locals,\nrecursive_guard)\r\n      328     if isinstance(t, (_GenericAlias, GenericAlias,\ntypes.UnionType)):\r\n--> 329         ev_args = tuple(_eval_type(a, globals,\nlocals, recursive_guard) for a in t.__args__)\r\n      330         if ev_args ==\nt.__args__:\r\nFile /opt/conda/lib/python3.10/typing.py:329, in\n<genexpr>(.0)\r\n      328     if isinstance(t, (_GenericAlias, GenericAlias,\ntypes.UnionType)):\r\n--> 329         ev_args = tuple(_eval_type(a, globals,\nlocals, recursive_guard) for a in t.__args__)\r\n      330         if ev_args ==\nt.__args__:\r\nFile /opt/conda/lib/python3.10/typing.py:327, in\n_eval_type(t, globals, locals, recursive_guard)\r\n      326     if isinstance(t,
```

```

ForwardRef):\r\n--> 327      return t._evaluate(globalns, localns,
recursive_guard)\r\n      328 if isinstance(t, (_GenericAlias, GenericAlias,
types.UnionType)):\r\n\r\nFile /opt/conda/lib/python3.10/typing.py:699, in
ForwardRef._evaluate(self, globalns, localns, recursive_guard)\r\n      693 type_
= _type_check(\r\n      694      eval(self.__forward_code__, globalns,
localns),\r\n      695      \"Forward references must evaluate to types.\",\r\n
696      is_argument=self.__forward_is_argument__,\r\n      697
allow_special_forms=self.__forward_is_class__,\r\n      698 )\r\n--> 699
self.__forward_value__ = _eval_type(\r\n      700      type_, globalns, localns,
recursive_guard | {self.__forward_arg__})\r\n      701 )\r\n      702
self.__forward_evaluated__ = True\r\n\r\nFile
/opt/conda/lib/python3.10/typing.py:329, in _eval_type(t, globalns, localns,
recursive_guard)\r\n      328 if isinstance(t, (_GenericAlias, GenericAlias,
types.UnionType)):\r\n--> 329      ev_args = tuple(_eval_type(a, globalns,
localns, recursive_guard) for a in t.__args__)\r\n      330      if ev_args ==
t.__args__:\r\n\r\nFile /opt/conda/lib/python3.10/typing.py:329, in
<genexpr>(.0)\r\n      328 if isinstance(t, (_GenericAlias, GenericAlias,
types.UnionType)):\r\n--> 329      ev_args = tuple(_eval_type(a, globalns,
localns, recursive_guard) for a in t.__args__)\r\n      330      if ev_args ==
t.__args__:\r\n\r\nFile /opt/conda/lib/python3.10/typing.py:329, in
_eval_type(t, globalns, localns, recursive_guard)\r\n      328 if isinstance(t,
(_GenericAlias, Generi",
    "GitHub_Issue_vector": [
        -0.0006835468229837716,
        -0.003785524284467101,
        0.006393408868461847,
        -0.007254402153193951,
        -0.005251347552984953,
        0.03227657452225685,
        -0.00281779607757926,
        -0.029031839221715927,
        -0.02083461359143257,
        -0.01790296845138073,
        0.024776682257652283,
        0.003934952896088362,
        -0.01551211066544056,
        0.02352432906627655,
        0.012701429426670074,
        -0.020436137914657593,
        0.015995973721146584,
        -0.019126858562231064,
        0.013199524953961372,
        0.006592646706849337,
        0.007001796737313271,
        0.025360165163874626,
        -0.02823488786816597,
        -0.004251598846167326,
        -0.014501688070595264,

```

0.011491769924759865,
0.024975920096039772,
-0.036830589175224304,
-0.010588082484900951,
-0.000776939676143229,
0.01770372875034809,
-0.022627757862210274,
-0.010424423031508923,
-0.03745676577091217,
-0.025687485933303833,
-0.009840940125286579,
0.018899157643318176,
-0.01787450537085533,
0.04628016799688339,
0.019069934263825417,
0.02086307667195797,
0.016437144950032234,
0.0004998741787858307,
-0.0170917846262455,
-0.02271314524114132,
0.04482857510447502,
-0.00794106163084507,
-0.004084381274878979,
-0.023296628147363663,
0.00932149775326252,
0.04129921644926071,
0.018884927034378052,
-0.027480628341436386,
-0.0025402859319001436,
0.00617638137191534,
-0.002337489975616336,
0.012829511426389217,
0.0290460716933012,
0.010004599578678608,
0.007592395413666964,
0.0075781638734042645,
-0.004685653373599052,
-0.018557608127593994,
0.00518374890089035,
-0.020635375753045082,
0.005799252074211836,
0.00018734164768829942,
0.02823488786816597,
-0.0013697620015591383,
0.004806619603186846,
0.022058505564928055,
0.005308272782713175,
0.002122241770848632,

-0.02641328051686287,
0.031650397926568985,
-0.02066383883357048,
-0.0015645528910681605,
0.0026061057578772306,
-0.019980736076831818,
0.009620354510843754,
0.013669157400727272,
-0.020165743306279182,
-0.014010708779096603,
0.0015014015370979905,
0.03532207012176514,
-0.0006742075202055275,
0.013718967325985432,
0.007542585954070091,
0.01874261349439621,
-0.015341334976255894,
0.030312655493617058,
0.0045967078767716885,
0.028419893234968185,
-0.0019888232927769423,
0.011698123998939991,
0.025630559772253036,
-0.006229748949408531,
0.012338532134890556,
0.007254402153193951,
-0.010652123019099236,
-0.012530654668807983,
0.005710306577384472,
-0.01999496854841709,
-0.014231293462216854,
-0.020720763131976128,
-0.02897491492331028,
0.012665851972997189,
0.021361172199249268,
0.008787823840975761,
-0.0003795753000304103,
-0.013740314170718193,
0.04895564913749695,
-0.007300653960555792,
-0.029971104115247726,
0.0020332960411906242,
-0.010652123019099236,
-0.003970530815422535,
-4.427822568686679e-05,
-0.009008409455418587,
-0.007535470183938742,
0.031223459169268608,

0.018244517967104912,
-0.018486451357603073,
-0.01398224662989378,
0.023851647973060608,
-0.004511320032179356,
-0.0083110760897398,
-0.002851595403626561,
-0.0323619619011879,
-0.017177172005176544,
0.023211240768432617,
-0.005699633155018091,
0.019710341468453407,
-0.0017958113458007574,
-0.025089770555496216,
0.028547976166009903,
-0.023268165066838264,
-0.01669330708682537,
-0.0046394020318984985,
-0.022200819104909897,
-0.001177639584057033,
0.04121382534503937,
-0.01468669530004263,
-0.002607884583994746,
0.013356069102883339,
-0.004991626366972923,
-0.02197311818599701,
-0.0025598539505153894,
0.04451548680663109,
-0.0012292279861867428,
0.004724789410829544,
0.0009517177822999656,
-0.010239415802061558,
-0.02147502265870571,
-0.029771866276860237,
0.014224178157746792,
-0.00775605533272028,
5.4229010856943205e-05,
-0.0025616330094635487,
-0.02900337614119053,
-0.0012701429659500718,
0.033130452036857605,
-0.01605289988219738,
-0.029743405058979988,
-0.015611729584634304,
0.017291022464632988,
0.01965341717004776,
-0.01770372875034809,
-0.0145230358466506,

0.0006404082523658872,
-0.03543592244386673,
0.03179271146655083,
-0.004369007423520088,
-0.0032447350677102804,
-0.0071512251161038876,
0.022627757862210274,
-0.006123014260083437,
-0.00256519066169858,
-0.02005189284682274,
0.010139796882867813,
-0.017177172005176544,
0.013640695251524448,
0.027067920193076134,
0.04263695701956749,
-0.010623660869896412,
0.009755551815032959,
0.024107811972498894,
0.007250844035297632,
-0.007563932798802853,
-0.006727844011038542,
0.005137497093528509,
0.03381355479359627,
-0.011079062707722187,
0.004543340764939785,
-0.5765382051467896,
-0.038282182067632675,
0.007286422420293093,
-0.007386041339486837,
-0.005258462857455015,
0.026100192219018936,
-0.010773089714348316,
0.02234313078224659,
-0.022300437092781067,
0.04337698221206665,
-0.05686825141310692,
0.011619850993156433,
0.017077552154660225,
0.002663030754774809,
0.014010708779096603,
-0.02002342976629734,
0.029401853680610657,
-0.03574901074171066,
-0.002260996727272868,
0.026270968839526176,
-0.005813483614474535,
0.02863336354494095,
-0.010794436559081078,

-0.013327606953680515,
0.00747142918407917,
0.0034599832724779844,
-0.0027644287329167128,
0.00465363310649991,
0.03398432955145836,
-0.002602547872811556,
-0.02917415276169777,
0.01696370169520378,
0.020848846063017845,
9.456028055865318e-05,
0.04730482026934624,
-0.014587076380848885,
-0.020535757765173912,
0.03742830455303192,
0.008602817542850971,
0.005873966496437788,
-0.012765470892190933,
0.012018327601253986,
0.01108617801219225,
0.018358368426561356,
0.00023214798420667648,
-0.016679076477885246,
0.006642456632107496,
-0.007176130078732967,
0.0002855153288692236,
7.532579184044152e-05,
0.021716954186558723,
-0.01521325297653675,
-0.013953783549368382,
0.005226442590355873,
0.003646769095212221,
-0.015170559287071228,
0.017006395384669304,
-0.016878314316272736,
0.021517716348171234,
-0.012302953749895096,
0.0008520987466908991,
-0.00011840881052194163,
-0.00804068148136139,
-0.016166750341653824,
0.01682139001786709,
0.03455357998609543,
0.03267505019903183,
0.02806411124765873,
-0.009691511280834675,
-0.04201078042387962,
0.013256450183689594,

-0.006236864719539881,
-0.03526514768600464,
0.006265327334403992,
0.02019420638680458,
0.037570614367723465,
0.0010130902519449592,
-0.005710306577384472,
-0.004696326795965433,
0.03028419427573681,
-0.005550204776227474,
0.0012203335063531995,
-0.04184000566601753,
-0.011207143776118755,
0.020436137914657593,
-0.016679076477885246,
-0.013498382642865181,
0.018087973818182945,
0.015597498044371605,
0.007528354413807392,
0.010154028423130512,
0.031109608709812164,
-0.008254150860011578,
-0.033528927713632584,
0.005792136769741774,
0.021859267726540565,
-0.003475993638858199,
-0.00844627246260643,
-0.0012292279861867428,
-0.05598590895533562,
-0.022641988471150398,
0.003959857393056154,
-0.0014827229315415025,
0.006859483662992716,
0.015384028665721416,
0.025417091324925423,
-0.005564435850828886,
-0.0020920000970363617,
0.01871415227651596,
-0.003438636427745223,
-0.007784517947584391,
-0.010068640112876892,
-0.011349457316100597,
-0.018600301817059517,
0.009342844597995281,
-0.04283619299530983,
-0.008958599530160427,
-0.018059512600302696,
-0.02052152529358864,

-0.0016214780043810606,
0.005105476826429367,
0.0263136625289917,
0.011228490620851517,
0.004440163727849722,
0.018685689195990562,
0.00569607550278306,
-0.0030739593785256147,
0.009534967131912708,
-0.000565249181818217,
0.030141880735754967,
0.021716954186558723,
-0.004774598870426416,
0.009115143679082394,
-0.017746424302458763,
0.004205347504466772,
-0.01142772939056158,
0.04021763801574707,
0.024406669661402702,
0.015640191733837128,
0.004682095721364021,
-0.015796735882759094,
-0.019568029791116714,
-2.2806203560321592e-05,
0.006386293098330498,
-0.030170343816280365,
-0.030682669952511787,
-0.021489253267645836,
-0.003767735091969371,
-0.011947170831263065,
-0.007727592717856169,
-0.0072686332277953625,
-0.0056604971177875996,
-0.02446359395980835,
0.021176164969801903,
-0.006681592669337988,
-0.012210450135171413,
0.00320915668271482,
-0.021830804646015167,
0.012317185290157795,
-0.026470206677913666,
0.010637892410159111,
0.03065420687198639,
-0.02281276322901249,
-0.009008409455418587,
-0.023382015526294708,
-0.0003789082111325115,
-0.02285545878112316,

0.016380218788981438,
-0.013903974555432796,
-0.028391432017087936,
-0.008851864375174046,
-0.0415838398039341,
-0.004187558311969042,
-0.0018998776795342565,
-0.002244986593723297,
0.00928591936826706,
-0.01992381177842617,
-0.0030810751486569643,
0.004603823646903038,
0.002892510499805212,
-0.015554804354906082,
-0.0020350751001387835,
-0.025459785014390945,
-0.016650613397359848,
0.04838639870285988,
0.014288218691945076,
0.04118536412715912,
-0.003547149943187833,
-0.0002071320341201499,
0.04149845242500305,
0.0031717997044324875,
-0.019624954089522362,
-0.008147415705025196,
0.004162653349339962,
-0.031308844685554504,
0.002182724652811885,
-0.006101667415350676,
0.015255946666002274,
-0.004881334025412798,
0.020436137914657593,
0.007563932798802853,
0.040530726313591,
0.024420900270342827,
-0.009940559044480324,
0.0018082638271152973,
-0.018287213519215584,
0.0023855206090956926,
-0.026142887771129608,
0.003091748571023345,
-0.0014133453369140625,
0.016465606167912483,
0.003413731697946787,
-0.04926874116063118,
-0.01807374320924282,
0.017106015235185623,

0.016906777396798134,
-0.03028419427573681,
0.014430532231926918,
-0.017575647681951523,
0.012544885277748108,
0.010239415802061558,
0.01844375766813755,
0.00986228697001934,
-0.013064327649772167,
-0.005358082242310047,
-0.007414503954350948,
0.026057498529553413,
-0.010744626633822918,
-0.008844749070703983,
-0.001042442279867828,
-0.005329619627445936,
-0.0013875511940568686,
0.011292532086372375,
-0.008538776077330112,
0.017319485545158386,
0.003372816601768136,
0.05023646727204323,
0.005422122776508331,
0.042181555181741714,
-0.028078343719244003,
0.002536728046834469,
-0.009414001367986202,
-0.013405879028141499,
-0.007670667488127947,
0.03307352587580681,
0.008346653543412685,
0.03213426098227501,
0.014871702529489994,
-0.014843239448964596,
0.011377919465303421,
-0.0028942893259227276,
-0.003902932396158576,
-0.01468669530004263,
-0.006094551645219326,
0.006806116085499525,
-0.022286206483840942,
0.010488463565707207,
0.023851647973060608,
0.007215266115963459,
0.0016570562729611993,
0.0076208580285310745,
-0.003564939135685563,
0.01965341717004776,

0.046792495995759964,
0.0103674978017807,
-0.014729388989508152,
-0.03785524144768715,
-0.0369444377720356,
-0.0038993745110929012,
-0.0037001364398747683,
-0.011385034769773483,
-0.026399049907922745,
-0.020649608224630356,
-0.026441743597388268,
0.05592898651957512,
-0.0038993745110929012,
0.015113634057343006,
0.04104305058717728,
0.014700926840305328,
-0.003991878125816584,
0.0028907316736876965,
-0.03028419427573681,
0.01908416487276554,
0.021859267726540565,
-0.007542585954070091,
-0.007233055308461189,
0.005329619627445936,
0.007136994041502476,
-0.02900337614119053,
0.01834413781762123,
0.008588586002588272,
0.012473729439079762,
-0.01928340271115303,
0.012857973575592041,
-0.022940846160054207,
-0.013761661015450954,
0.013256450183689594,
0.002156041096895933,
0.0015814525540918112,
0.015469416044652462,
-0.0013964456738904119,
0.027338314801454544,
0.010943865403532982,
0.03455357998609543,
0.04067303612828255,
-0.012423919513821602,
-0.011996980756521225,
-0.008837633766233921,
-0.01495708990842104,
-0.040701501071453094,
0.032191187143325806,

0.013583770021796227,
-0.005692517384886742,
-0.025986341759562492,
-0.0013822143664583564,
0.024477824568748474,
-0.009705742821097374,
-0.007016027811914682,
0.007293538190424442,
-0.021845035254955292,
-0.01618098095059395,
-0.042722344398498535,
-0.014039171859622002,
-0.0018322791438549757,
0.014658233150839806,
0.023467402905225754,
0.012103715911507607,
0.003337238449603319,
-0.0007351352833211422,
0.014024940319359303,
0.004820850677788258,
0.004739020951092243,
-0.00206531654112041,
-0.022698914632201195,
-0.03176424652338028,
0.00797664001584053,
0.002280564745888114,
0.02796449325978756,
0.03065420687198639,
0.016949471086263657,
0.022556601092219353,
-0.01955379731953144,
0.008126068860292435,
-0.021105008199810982,
0.02355279214680195,
-0.016465606167912483,
0.017646804451942444,
0.02789333648979664,
0.008937252685427666,
0.008553007617592812,
0.014629770070314407,
0.007421619724482298,
0.010595198720693588,
-0.02197311818599701,
-0.007997986860573292,
0.017618341371417046,
0.007692014332860708,
0.018500681966543198,
-0.011755048297345638,

0.032760437577962875,
0.008111837320029736,
0.005326061509549618,
-0.006112340837717056,
0.0113067626953125,
0.04556860402226448,
0.029658015817403793,
-6.320695683825761e-05,
-0.028761444613337517,
0.003963415510952473,
-0.021375402808189392,
0.003657442517578602,
-0.01935455948114395,
-0.01645137555897236,
-0.026882914826273918,
0.03677366301417351,
0.013818586245179176,
-0.02665521390736103,
-0.03640364855527878,
0.02594364807009697,
-0.006432544905692339,
-0.020621145144104958,
-0.017305253073573112,
-0.015312871895730495,
-0.007186803501099348,
-0.00908668152987957,
-0.025730179622769356,
-0.013832817785441875,
-0.007247286383062601,
0.0009775119833648205,
0.005863293074071407,
0.0009970800019800663,
0.018870696425437927,
-0.02453475072979927,
-0.013327606953680515,
0.026370586827397346,
0.0108371302485466,
0.0004149311571381986,
-0.010054409503936768,
0.010901170782744884,
-0.01837260089814663,
0.026740601286292076,
-0.03338661417365074,
-0.0023446057457476854,
0.02843412570655346,
5.395800599217182e-06,
-0.03836756944656372,
-0.018699919804930687,

-0.024520520120859146,
-0.008204340934753418,
-0.016038667410612106,
0.018258750438690186,
-0.01945417933166027,
0.006525048054754734,
0.04209616780281067,
0.011420613154768944,
0.013320490717887878,
0.00023437161871697754,
-0.018130667507648468,
0.010573850944638252,
-0.004443721380084753,
-0.024178968742489815,
0.047333285212516785,
-0.017077552154660225,
0.012345647439360619,
0.012850858271121979,
-0.022428520023822784,
-0.020179973915219307,
-0.029999567195773125,
0.010068640112876892,
-0.01955379731953144,
-0.023040464147925377,
0.007563932798802853,
0.0011135987006127834,
0.002211187267675996,
0.006891503930091858,
-0.023168547078967094,
8.160757715813816e-05,
-0.01766103506088257,
-0.0024050886277109385,
-0.007314885035157204,
0.030967295169830322,
0.008631279692053795,
-0.017447566613554955,
-0.008424925617873669,
0.003397721331566572,
-0.04004685953259468,
0.03025573119521141,
0.02493322640657425,
-0.007663551717996597,
0.02335355244576931,
0.016550995409488678,
-0.025801336392760277,
-0.005578667391091585,
-0.015312871895730495,
0.01938302256166935,

0.01162696722894907,
0.008396463468670845,
-0.03176424652338028,
0.0028338064439594746,
-0.020080355927348137,
-0.012381225824356079,
-0.010680586099624634,
-0.01216775644570589,
-0.009890749119222164,
-0.013050096109509468,
0.018429525196552277,
-0.015042477287352085,
-0.0202653631567955,
-0.004095054697245359,
-0.046621717512607574,
-0.022001581266522408,
0.01006152480840683,
0.022428520023822784,
0.03287428990006447,
-0.0032127145677804947,
0.019624954089522362,
-0.008652626536786556,
-0.005699633155018091,
-0.025872493162751198,
0.003796197706833482,
0.011854668147861958,
0.02658405713737011,
0.017319485545158386,
0.0230831578373909,
0.059145256876945496,
-0.026897145435214043,
-0.002488697413355112,
0.011449076235294342,
0.01425264123827219,
0.00594868091866374,
-0.012281606905162334,
0.009563429281115532,
-0.015896355733275414,
0.006503701210021973,
0.004956047981977463,
0.0008854533080011606,
0.02668367512524128,
0.0045682452619075775,
-0.014010708779096603,
0.036289799958467484,
-0.001310168532654643,
0.011022137477993965,
-0.026882914826273918,

-0.04445856064558029,
-0.043633148074150085,
0.02742370404303074,
-0.02436397597193718,
0.013299143873155117,
-0.03768446668982506,
0.0009277024655602872,
0.0387660451233387,
-0.013491266407072544,
-0.0019087722757831216,
0.004184000194072723,
-0.0025047077797353268,
-0.01723409630358219,
0.018045280128717422,
0.006350714713335037,
0.030881907790899277,
-0.001238122582435608,
-0.02160310372710228,
-0.013327606953680515,
-0.016935240477323532,
-0.005567993503063917,
-0.012893551960587502,
0.02382318675518036,
0.006489470135420561,
0.02746639773249626,
-0.029102995991706848,
0.05550204589962959,
0.011114640161395073,
0.01955379731953144,
0.008318191394209862,
-0.010822898708283901,
-0.024520520120859146,
-0.04166922718286514,
-0.02174541726708412,
-0.02362394705414772,
0.025388628244400024,
0.0030775172635912895,
0.009414001367986202,
0.019966505467891693,
-0.0212046280503273,
-0.027409471571445465,
0.006610435899347067,
-0.018358368426561356,
0.023410478606820107,
0.022086968645453453,
0.037542153149843216,
0.02934492751955986,
-0.0012034338433295488,

-0.01871415227651596,
0.013149715960025787,
0.01787450537085533,
-0.007322000805288553,
0.0029263098258525133,
-0.010161143727600574,
-0.01891339011490345,
-0.0017646804917603731,
0.009250340983271599,
0.010481348261237144,
-0.026470206677913666,
-0.03153654560446739,
0.03640364855527878,
0.019639184698462486,
0.011000790633261204,
-0.015298641286790371,
-0.017205635085701942,
-0.021076546981930733,
0.01208236813545227,
0.006937755737453699,
-0.01415302138775587,
0.0258155670017004,
-0.00373927247710526,
0.002440667012706399,
0.011207143776118755,
-0.01494285836815834,
0.057864442467689514,
0.02197311818599701,
0.006507259327918291,
0.03492359444499016,
-0.007464313413947821,
0.0005794804892502725,
0.024876302108168602,
0.010545388795435429,
0.03546438366174698,
-0.018415294587612152,
-0.013676273636519909,
0.024207431823015213,
-0.007983756251633167,
0.0066958237439394,
-0.006382735446095467,
0.010716164484620094,
0.03532207012176514,
-0.010303456336259842,
0.0015245273243635893,
-0.007627973333001137,
-0.016977934166789055,
0.014985552057623863,

0.0016846293583512306,
0.00751412333920598,
-0.0030294866301119328,
-0.02443513087928295,
-0.00489556510001421,
0.01420283131301403,
0.03722906485199928,
0.008617048151791096,
-0.010467116720974445,
0.002412204397842288,
-0.01773219183087349,
0.008723783306777477,
-0.011925823986530304,
0.004625170491635799,
-0.046564795076847076,
-0.006955544929951429,
-0.0009650596184656024,
-0.02005189284682274,
0.0017806906253099442,
-0.004863544832915068,
-0.0007475876482203603,
-0.02867605723440647,
0.035350535064935684,
-0.007457198109477758,
0.0414699912071228,
-0.032931212335824966,
0.007763170637190342,
-0.028377199545502663,
-0.008403578773140907,
-0.0006835468229837716,
0.0013341838493943214,
0.010737511329352856,
0.006799000781029463,
-0.021901961416006088,
-0.028277581557631493,
0.03685905039310455,
0.019098395481705666,
-0.003573833731934428,
-0.017106015235185623,
0.03321583941578865,
0.0035756125580519438,
0.015156327746808529,
-0.011264069005846977,
0.0007364694611169398,
0.05373736470937729,
-0.008581469766795635,
0.0023357111494988203,
0.00694842915982008,

-0.017589880153536797,
-0.01608136296272278,
-0.01689254492521286,
0.005841946229338646,
-0.00962747074663639,
-0.022898152470588684,
0.012338532134890556,
-0.004803061485290527,
-0.021958885714411736,
-0.002321479842066765,
-0.022030042484402657,
-0.01716293953359127,
-0.03856680542230606,
-0.029131459072232246,
0.001963918562978506,
-0.007322000805288553,
-0.02934492751955986,
0.020137280225753784,
0.021418096497654915,
0.0179741233587265,
0.004810177255421877,
-0.028362968936562538,
-0.006855926010757685,
-0.004201789386570454,
0.0001722208980936557,
-0.02463437058031559,
-0.007571048568934202,
-0.0030686226673424244,
0.03751369193196297,
0.032760437577962875,
0.002237871056422591,
-0.027551785111427307,
-0.024790914729237556,
-0.014999783597886562,
-0.018970314413309097,
-0.0010700153652578592,
-0.0003404392336960882,
0.02352432906627655,
-0.004945374559611082,
-0.032020412385463715,
0.018258750438690186,
0.006034068763256073,
0.015668654814362526,
-0.008424925617873669,
-0.028448356315493584,
0.0016063572838902473,
0.015255946666002274,
0.01396801508963108,

-0.0018500682199373841,
-0.020578451454639435,
-0.006396966520696878,
0.004297850653529167,
-0.0030810751486569643,
-0.004308524075895548,
0.011214260011911392,
-0.034980520606040955,
-0.007019585929811001,
-0.017618341371417046,
-0.011641198769211769,
0.0020546431187540293,
0.0029067418072372675,
0.000820967776235193,
-0.03364278003573418,
0.007848558947443962,
0.005856177303940058,
-0.007122762501239777,
0.002880058018490672,
-0.016750233247876167,
-0.003038381226360798,
0.007257959805428982,
0.02870452031493187,
-0.01495708990842104,
-0.008880327455699444,
-0.009584776125848293,
-0.030739594250917435,
0.049354128539562225,
0.021275784820318222,
0.005109034478664398,
0.00193901383318007,
-0.0014507025480270386,
0.012238912284374237,
-0.026142887771129608,
0.01056673564016819,
-0.011363687925040722,
-0.011854668147861958,
0.003499119309708476,
-0.01925494149327278,
0.012623157352209091,
-0.027082152664661407,
0.017504490911960602,
-0.025587866082787514,
-0.0018571838736534119,
0.01549787912517786,
0.0002299243351444602,
-0.02493322640657425,
0.02093423344194889,

-0.015526341274380684,
-0.04838639870285988,
-0.01108617801219225,
0.004486415535211563,
0.010637892410159111,
0.02991417981684208,
-0.022015811875462532,
0.006187054794281721,
-0.023382015526294708,
-0.01854337565600872,
0.015440953895449638,
0.004048802889883518,
-0.009463810361921787,
0.007948177866637707,
0.0015174116706475616,
-0.02177388034760952,
-0.019610723480582237,
0.20743533968925476,
0.010794436559081078,
-0.004682095721364021,
0.01366204209625721,
-0.002652357332408428,
0.00880205538123846,
0.001841173623688519,
-0.0038887010887265205,
0.004874218255281448,
0.037001363933086395,
0.0063400412909686565,
0.0028818370774388313,
0.022528138011693954,
-0.009172068908810616,
-0.011776396073400974,
0.041413065046072006,
-0.029401853680610657,
-0.0488702617585659,
-0.04007532447576523,
-0.0028338064439594746,
-0.0011198249412700534,
-0.0009108028025366366,
0.0009615017916075885,
-0.02258506417274475,
0.017390640452504158,
-0.007599510718137026,
0.003182473126798868,
-0.014409185387194157,
-0.005044993478804827,
-0.006503701210021973,
-0.026427512988448143,

0.0019585818517953157,
-0.0004811956314370036,
-0.0017095342045649886,
-0.02870452031493187,
-0.030682669952511787,
0.00616570794954896,
0.0012532433029264212,
0.011135987006127834,
0.016679076477885246,
0.011940055526793003,
-0.017547184601426125,
0.011491769924759865,
-0.015967512503266335,
-0.02682598866522312,
0.004909796174615622,
-0.005859735421836376,
0.003442194312810898,
-0.010737511329352856,
0.020080355927348137,
-0.026342125609517097,
-0.020478831604123116,
0.01736217923462391,
0.016736000776290894,
0.0045682452619075775,
-0.02029382437467575,
0.006066089030355215,
0.012061021290719509,
-0.0029512145556509495,
-0.006610435899347067,
-0.029231077060103416,
0.009150722064077854,
-0.005137497093528509,
0.021318478509783745,
-0.036830589175224304,
0.00514817051589489,
-0.0034190684091299772,
0.006884388625621796,
0.007464313413947821,
-0.02254236862063408,
-0.01723409630358219,
-0.030711133033037186,
0.006834578700363636,
0.010310572572052479,
-0.028249118477106094,
-0.0262851994484663,
-0.0037001364398747683,
-0.01061654556542635,
-0.003998993430286646,

0.03284582495689392,
0.0011509559117257595,
0.006375619675964117,
-0.0023072485346347094,
0.010794436559081078,
-0.02759447880089283,
-0.017376409843564034,
0.0434054471552372,
0.005240674130618572,
0.01078732032328844,
-0.013519729487597942,
-0.02056421898305416,
0.011683892458677292,
-0.008979946374893188,
0.02029382437467575,
-0.007763170637190342,
0.026199812069535255,
0.025331703945994377,
0.0031700206454843283,
0.010182490572333336,
0.005101918708533049,
-0.03307352587580681,
0.0341266430914402,
0.02258506417274475,
-0.0023730683606117964,
-0.021958885714411736,
-0.008375116623938084,
0.00824703462421894,
-0.0017344389343634248,
-0.02207273617386818,
-0.03560669720172882,
-0.020549988374114037,
-0.030682669952511787,
0.0011376140173524618,
0.013548191636800766,
0.003974088933318853,
-0.007464313413947821,
0.004539782647043467,
-0.017020627856254578,
0.022001581266522408,
0.007197476923465729,
0.02880413830280304,
-0.026840221136808395,
-0.006090993992984295,
0.029231077060103416,
-0.0026345683727413416,
-0.024876302108168602,
-0.01736217923462391,

0.012978940270841122,
-0.009734204970300198,
-0.017390640452504158,
0.027437934651970863,
-0.02126155234873295,
0.006080320104956627,
0.0018233845476061106,
-0.023282397538423538,
0.021816574037075043,
-0.0026096634101122618,
-0.008908789604902267,
0.015825198963284492,
-0.006012721452862024,
0.007869905792176723,
0.011043484322726727,
0.014124559238553047,
-0.03187809884548187,
0.004304966423660517,
-0.03896528109908104,
0.0021115681156516075,
-0.009549197740852833,
-0.015995973721146584,
-0.024591675028204918,
-0.018087973818182945,
-0.003047275822609663,
0.004233810119330883,
-0.013235103338956833,
0.013420110568404198,
-0.014394953846931458,
-0.0006595315062440932,
-0.036432113498449326,
-0.029572628438472748,
0.013178178109228611,
-0.03284582495689392,
0.018130667507648468,
0.014288218691945076,
-0.010936749167740345,
-0.026569826528429985,
-0.03725752606987953,
-0.18159131705760956,
0.014985552057623863,
0.025872493162751198,
-0.03819679096341133,
0.01295759342610836,
0.030739594250917435,
0.03119499608874321,
-0.004493530839681625,
-0.015298641286790371,

-0.02133270911872387,
0.04852871224284172,
0.0054078917019069195,
-0.017006395384669304,
-0.01208236813545227,
-0.017789117991924286,
0.0052762520499527454,
-0.035065907984972,
-0.012850858271121979,
0.0332443043589592,
0.03341507911682129,
0.033699702471494675,
-0.008097605779767036,
0.02278430201113224,
0.025417091324925423,
0.0037250411696732044,
0.010630776174366474,
-0.021218858659267426,
0.020236900076270103,
0.0037143677473068237,
-0.021290015429258347,
0.01992381177842617,
-0.0052762520499527454,
0.03196348622441292,
-0.005012973211705685,
0.02645597606897354,
0.008517429232597351,
0.02312585338950157,
0.0017522280104458332,
-0.013199524953961372,
0.0103674978017807,
-0.0022787859197705984,
0.01598174311220646,
0.005208653397858143,
-0.024691294878721237,
-0.031678859144449234,
0.00955631397664547,
0.002746639773249626,
-0.018756845965981483,
0.02776525355875492,
-0.007229497190564871,
0.014373607002198696,
-0.024847839027643204,
-0.013085674494504929,
-0.002647020621225238,
-0.004162653349339962,
0.02281276322901249,
-0.010751742869615555,

0.0013581991661339998,
-0.0005314498557709157,
0.006599762476980686,
0.009342844597995281,
-0.007044490426778793,
0.021375402808189392,
-0.015355566516518593,
-0.008467620238661766,
-0.00021658249897882342,
-0.022200819104909897,
0.024107811972498894,
-0.009043986909091473,
0.015398260205984116,
-0.009534967131912708,
0.009840940125286579,
6.331814074655995e-05,
0.003027707803994417,
0.002684377832338214,
0.01341299433261156,
-0.013676273636519909,
0.00466074887663126,
-0.00037179255741648376,
-0.009293034672737122,
-0.003394163679331541,
0.016024436801671982,
-0.002940541133284569,
-0.005994932726025581,
0.015170559287071228,
-0.0027324084658175707,
0.012032559141516685,
0.0010655681835487485,
-0.03557823598384857,
-0.01189736183732748,
0.013484151102602482,
-0.00284803775139153,
-0.008745130151510239,
-0.026769064366817474,
0.004746136721223593,
0.013178178109228611,
-0.01972457394003868,
-0.01861453242599964,
0.00013330719957593828,
-0.012943361885845661,
0.00233037443831563,
0.0022521023638546467,
-0.010758858174085617,
0.010879823938012123,
0.00047096688649617136,

0.018685689195990562,
0.006976891774684191,
0.037542153149843216,
0.03341507911682129,
-0.023766260594129562,
0.020535757765173912,
0.025459785014390945,
0.022826995700597763,
0.013092790730297565,
-0.02288391999900341,
0.021517716348171234,
-0.012196218594908714,
-0.028761444613337517,
-0.00947092566639185,
0.0024869185872375965,
0.06927794218063354,
0.011982749216258526,
0.035350535064935684,
0.011299647390842438,
-0.005044993478804827,
-0.020350750535726547,
-0.08533083647489548,
-0.043690070509910583,
0.0033603643532842398,
0.007919714786112309,
-0.0263136625289917,
-0.004703442566096783,
-0.013590885326266289,
0.038310643285512924,
-0.020137280225753784,
-0.0018803096609190106,
0.00670649716630578,
0.00873089861124754,
0.002150704385712743,
-0.002702167024835944,
-0.0069875651970505714,
-0.013028749264776707,
-0.0056604971177875996,
0.007247286383062601,
-0.011833321303129196,
0.01605289988219738,
-0.01396801508963108,
-0.011683892458677292,
0.006937755737453699,
-0.008609932847321033,
0.007485660724341869,
-0.002293017227202654,
-0.037570614367723465,

-0.009634586051106453,
-0.004098612815141678,
0.017205635085701942,
0.019738804548978806,
-0.01635175757110119,
-0.012139293365180492,
-0.014053402468562126,
-0.02086307667195797,
-0.02382318675518036,
-0.004262272734194994,
-0.00647879671305418,
0.02012304961681366,
-0.010225184261798859,
0.027366777881979942,
0.02278430201113224,
0.03025573119521141,
-0.0011678555747494102,
-0.008673973381519318,
-0.007300653960555792,
-0.003511571791023016,
0.040900737047195435,
-0.009485157206654549,
-0.017376409843564034,
-0.03270351514220238,
-0.022841226309537888,
-0.011221375316381454,
-0.0033550276421010494,
0.0230831578373909,
-0.0056569394655525684,
0.014053402468562126,
0.01783181168138981,
0.012950477190315723,
-0.013996477238833904,
0.011463306844234467,
-0.013078559190034866,
-0.005005857441574335,
-0.000877892947755754,
0.005731653422117233,
0.015540572814643383,
-0.010943865403532982,
-0.014466110616922379,
0.03785524144768715,
-0.03372816741466522,
0.003838891629129648,
-0.00935707613825798,
0.001424018875695765,
0.02194465510547161,
-0.029971104115247726,

-0.017106015235185623,
0.016010206192731857,
-0.01834413781762123,
0.011328109540045261,
-0.01240257266908884,
-0.035919785499572754,
0.009065333753824234,
0.019411485642194748,
0.004159095697104931,
0.03819679096341133,
0.02396549843251705,
0.006624667439609766,
0.014003593474626541,
0.010068640112876892,
-0.02604326792061329,
0.030739594250917435,
0.00016810715897008777,
0.026939839124679565,
-0.012502191588282585,
-0.014046287164092064,
0.013690505176782608,
0.0029102996923029423,
-0.016977934166789055,
-0.027352547273039818,
0.007848558947443962,
-0.039221446961164474,
-0.025801336392760277,
-0.0756535604596138,
0.028476819396018982,
0.010943865403532982,
-0.01494285836815834,
-0.011235606856644154,
-0.05026492848992348,
0.010908287018537521,
-0.011221375316381454,
-0.019938042387366295,
0.020478831604123116,
-0.008837633766233921,
0.013434341177344322,
0.01494285836815834,
0.0036396533250808716,
0.00217916676774621,
-0.0007698240806348622,
0.022186586633324623,
-0.0011233827099204063,
0.021318478509783745,
-0.008432041853666306,
0.007869905792176723,

0.012210450135171413,
0.021716954186558723,
0.023296628147363663,
-0.0035133506171405315,
0.011313878931105137,
0.009442463517189026,
0.010040177963674068,
-0.010659239254891872,
-0.017404872924089432,
0.01468669530004263,
-0.013718967325985432,
0.034781280905008316,
0.025957880541682243,
-0.011235606856644154,
0.005358082242310047,
0.0470486581325531,
0.0008325306698679924,
0.05313965305685997,
0.007069395389407873,
-0.008937252685427666,
-0.0434054471552372,
0.02776525355875492,
-0.03179271146655083,
-0.013825701549649239,
-0.024719757959246635,
-0.022698914632201195,
0.0020670953672379255,
0.03278890252113342,
0.018970314413309097,
0.01142772939056158,
0.002883615903556347,
-0.04408854991197586,
-0.030568819493055344,
-0.003385269083082676,
-0.02386588044464588,
0.026939839124679565,
0.011918708682060242,
0.017760654911398888,
-0.0020688744261860847,
0.029800329357385635,
-0.013569538481533527,
0.012594695203006268,
-0.007012470159679651,
0.006806116085499525,
-0.018429525196552277,
-0.02083461359143257,
0.01443764753639698,
-0.003009918611496687,

-0.025858260691165924,
-0.033130452036857605,
-0.012089484371244907,
0.015270178206264973,
0.012174871750175953,
0.01396801508963108,
0.014772082678973675,
0.0030152553226798773,
0.013811470940709114,
-0.006293789949268103,
0.02046460099518299,
0.02769409865140915,
-0.0005412339232861996,
-0.002627452602609992,
0.01263738889247179,
0.04747559502720833,
-0.001073573250323534,
-0.028932221233844757,
-0.0026701465249061584,
-0.02086307667195797,
-0.031109608709812164,
-0.027850642800331116,
-0.007635089103132486,
-0.003584507154300809,
0.007507007569074631,
-0.0005221106112003326,
0.018287213519215584,
-0.016408681869506836,
0.009784014895558357,
0.0026790411211550236,
-0.005585782695561647,
0.02907453291118145,
0.006311578676104546,
0.009349959902465343,
-0.001500512007623911,
-0.028960682451725006,
-0.01186178345233202,
-0.03691597655415535,
-0.030312655493617058,
-0.01468669530004263,
-0.019738804548978806,
-0.006905735470354557,
0.0022876805160194635,
-0.0015174116706475616,
0.017589880153536797,
-0.008346653543412685,
-0.005180190782994032,
-0.009620354510843754,

-0.02779371663928032,
-0.01393243670463562,
0.023908574134111404,
-0.0030294866301119328,
0.00439035426825285,
0.043092358857393265,
-0.017490260303020477,
0.04414547234773636,
0.034212030470371246,
-0.007542585954070091,
-0.00019323428568895906,
-0.0019425716018304229,
-0.00373927247710526,
0.001342188916169107,
-0.004667864181101322,
-0.021190397441387177,
-0.008282613009214401,
-0.01918378472328186,
-0.022613525390625,
-0.002908520633354783,
-0.006123014260083437,
-0.019667647778987885,
0.07468582689762115,
-0.017447566613554955,
-0.0014000035589560866,
-0.0184722188860178,
-0.010687701404094696,
0.029430314898490906,
0.02503284625709057,
0.016209444031119347,
-0.0021435886155813932,
-0.03919298201799393,
-0.014800545759499073,
-0.01884223334491253,
-0.004176884889602661,
-0.030597282573580742,
-0.02968647889792919,
0.013199524953961372,
-0.003764177206903696,
0.03409818187355995,
-0.0002546067407820374,
0.005272694397717714,
0.0020706532523036003,
0.019980736076831818,
0.027437934651970863,
0.019240709021687508,
-0.006144361104816198,
0.0013911089627072215,

```

        0.014558613300323486,
        0.02241428755223751,
        -0.010929633863270283,
        -0.03253273665904999,
        0.007012470159679651,
        0.002481581876054406,
        -0.011918708682060242,
        -0.02453475072979927,
        -0.006471680942922831,
        -0.0007440298213623464,
        0.000239930726820603,
        -0.0020688744261860847,
        0.01736217923462391,
        -0.020706532523036003,
        0.027110615745186806,
        0.025289008393883705,
        -0.003927837125957012,
        -0.0479879230260849,
        -0.005457701161503792,
        0.005386544857174158,
        -0.00435833353549242,
        -0.03919298201799393,
        -0.034952059388160706
    ],
}
},
{
    "_index": "github_issues",
    "_id": "CGLQUZMBKaR7I-arfsTE",
    "_score": 1.0,
    "_source": {
        "_type": "issue",
        "_repo": "langchain",
        "_issueNumber": "28270",
        "_title": "core: add space at end of error message link",
        "_createdAt": "2024-11-21T22:13:37Z",
        "_closedAt": "2024-11-21T22:19:59Z",
        "_state": "closed",
        "_body": "None",
        "GitHub_Issue_vector": [
            -0.014768942259252071,
            0.0032411953434348106,
            0.00551387807354331,
            -0.015006699599325657,
            -0.023342201486229897,
            0.008251586928963661,
            -0.008964859880506992,
            -0.03082457184791565,

```

-0.017188474535942078,
0.015817873179912567,
0.01241234503686428,
-0.012950795702636242,
-0.007936907932162285,
0.010321476496756077,
-0.00702084181830287,
-0.0004820709873456508,
0.012307452037930489,
-0.018307333812117577,
0.01267108041793108,
0.012153608724474907,
0.00758726429194212,
0.034796521067619324,
-0.010657134465873241,
0.0134053323417902,
0.003748178482055664,
0.015622071921825409,
0.013034709729254246,
-0.015915771946310997,
0.004898505751043558,
0.008412422612309456,
0.012936810031533241,
-0.00035488815046846867,
0.008251586928963661,
-0.016489187255501747,
-0.0037097176536917686,
0.010272526182234287,
0.015538157895207405,
-0.015664029866456985,
0.028446996584534645,
-0.022419141605496407,
0.011160620488226414,
0.0211604256182909,
0.012461294420063496,
-0.025356147438287735,
-0.03225111588835716,
0.016489187255501747,
0.0016686737071722746,
-0.02075483836233616,
-0.0064788940362632275,
0.02157999761402607,
0.0319993756711483,
-0.007412442471832037,
-0.025985505431890488,
0.016139544546604156,
0.005017384421080351,
0.006265611853450537,

0.00462927995249629,
0.02212544158101082,
0.001353120431303978,
-0.015580114908516407,
-0.0020978611428290606,
-0.01267108041793108,
-0.026670806109905243,
0.005555835552513599,
-0.022419141605496407,
-0.004660747945308685,
-0.019594023004174232,
0.015132571570575237,
0.0222513135522604,
-0.03311823308467865,
0.022545013576745987,
0.013999726623296738,
0.010083719156682491,
-0.013985740020871162,
0.020153451710939407,
-0.0016337092965841293,
0.014363355003297329,
-0.0013574908953160048,
0.020545052364468575,
0.010608184151351452,
0.01963597908616066,
-0.003561119083315134,
-0.03885238617658615,
0.0342930369079113,
0.03935587406158447,
-0.0012919327709823847,
0.002730715787038207,
0.015608086250722408,
0.0028740696143358946,
-0.01836327649652958,
0.02355198748409748,
0.022768786177039146,
0.03387346491217613,
0.02128629758954048,
0.013384353369474411,
0.02995745651423931,
-0.016167515888810158,
0.03700627014040947,
0.010622169822454453,
-0.04338376596570015,
-0.002131077228114009,
0.005615274887531996,
-0.014824884943664074,
-0.005797089543193579,

-0.0068530128337442875,
-0.016237445175647736,
0.0043285866267979145,
0.0026887585408985615,
0.014076647348701954,
0.012706045061349869,
-0.00759425712749362,
0.04727180302143097,
0.0022744310554116964,
-0.030069341883063316,
-0.007216642145067453,
-0.011803965084254742,
0.013824904337525368,
-0.03216720372438431,
0.005318077746778727,
-0.013747982680797577,
0.04162156209349632,
0.022293269634246826,
-0.004433479625731707,
-0.01297177467495203,
0.005174723919481039,
-0.002837357111275196,
-0.01354518998414278,
-0.008440393954515457,
0.0032202168367803097,
0.0035471334122121334,
0.047131944447755814,
0.022545013576745987,
0.006157222203910351,
0.012873874045908451,
-0.0017744408687576652,
0.007059302646666765,
-0.013580153696238995,
0.008776051923632622,
-0.016950717195868492,
-0.026684792712330818,
0.005699189379811287,
0.034628693014383316,
-0.02482469007372856,
-0.002423029625788331,
-0.008216622285544872,
0.024936575442552567,
-0.021719854325056076,
0.0027237229514867067,
0.028251195326447487,
-0.006230647210031748,
0.014573141932487488,
-0.024586930871009827,

-0.005363531410694122,
-0.014531183987855911,
-0.004398515447974205,
0.01813950575888157,
-0.023440100252628326,
0.0030838558450341225,
-0.020768824964761734,
-0.006734134163707495,
0.01518851425498724,
0.022447112947702408,
0.028377067297697067,
-0.01762203313410282,
0.0044159977696835995,
0.04858646169304848,
-0.001562906545586884,
-0.018279362469911575,
-0.013594139367341995,
-0.014782927930355072,
-0.03728598356246948,
-0.003954468294978142,
-0.02260095626115799,
0.02646102011203766,
0.0032429436687380075,
0.016852816566824913,
0.0016739183338358998,
0.030209198594093323,
-0.006898466497659683,
-0.005996386054903269,
-0.04388725385069847,
-0.0049579450860619545,
0.0007875720039010048,
0.03432100638747215,
-0.007699150126427412,
0.005401992239058018,
0.03292243182659149,
-0.014349369332194328,
0.030097313225269318,
0.0024090437218546867,
0.0245449747890234,
0.014531183987855911,
-0.01169907208532095,
0.013202538713812828,
-0.6086594462394714,
-0.02464287541806698,
-0.01032846886664629,
0.012055708095431328,
-0.0008299662731587887,
0.037090182304382324,

-0.005167731083929539,
0.004968434106558561,
-0.012629123404622078,
0.020600995048880577,
-0.03924398869276047,
0.023118428885936737,
0.00690196268260479,
-0.008391444571316242,
-0.0012062700698152184,
0.012789960019290447,
0.038964271545410156,
0.0031555327586829662,
-0.014391327276825905,
0.02364988625049591,
-0.03627901151776314,
0.03264271840453148,
0.03633495420217514,
-0.009055767208337784,
-0.011335442773997784,
-0.02750995196402073,
0.044642481952905655,
-0.008412422612309456,
-0.01120257843285799,
0.013510225340723991,
-0.017705947160720825,
-0.0028758179396390915,
0.01005574781447649,
-0.00027709247660823166,
0.053145814687013626,
-0.013594139367341995,
-0.04688020050525665,
0.008531301282346249,
0.004618790931999683,
0.02071288228034973,
-0.03191545978188515,
-0.0036572711542248726,
0.03356577828526497,
-0.0065767946653068066,
0.008433401584625244,
-0.035719580948352814,
0.005237659905105829,
-0.022069498896598816,
0.0068739913403987885,
0.014475241303443909,
0.012454302050173283,
-0.015552143566310406,
-0.013034709729254246,
0.023803729563951492,

-0.0030471431091427803,
-0.010824963450431824,
0.02706240862607956,
-0.0256638340651989,
0.019747866317629814,
-0.0011669351952150464,
-0.001495600095950067,
0.007069791667163372,
-0.007006856147199869,
-0.017412247136235237,
0.009538275189697742,
-0.00811172928661108,
0.005108291748911142,
-0.00040646057459525764,
0.017873777076601982,
-0.05823662504553795,
0.002026184229180217,
-0.007797050289809704,
-0.02820923924446106,
-0.020041566342115402,
-0.009083738550543785,
0.03502029553055763,
0.01686680316925049,
0.016838831827044487,
-0.02444707415997982,
0.00468522310256958,
-0.022684870287775993,
-0.025831662118434906,
-0.01816747710108757,
-0.01045434083789587,
0.023859674111008644,
0.012866880744695663,
-0.01107670646160841,
0.010370426811277866,
-0.023663872852921486,
-0.009006816893815994,
0.02556593343615532,
0.0017228684155270457,
-0.018726905807852745,
-0.011405371129512787,
-0.0033425919245928526,
0.01727238856256008,
0.016950717195868492,
0.008901923894882202,
0.016055630519986153,
-0.009776032529771328,
-0.005583806894719601,
0.012482273392379284,

-0.009636174887418747,
-0.005632756743580103,
0.02823721058666706,
-0.0029842073563486338,
-0.019552065059542656,
-0.018796835094690323,
0.016726946458220482,
0.003832092974334955,
-0.00379712856374681,
0.0047236839309334755,
0.0025069438852369785,
0.019454164430499077,
-0.0022691863123327494,
-0.03885238617658615,
0.016810860484838486,
-0.007727121468633413,
-0.003919503651559353,
0.02250305563211441,
0.001570773427374661,
0.01632135920226574,
-0.010692098177969456,
0.017076589167118073,
0.02900642529129982,
0.005929953884333372,
0.0079718716442585,
-0.03695032745599747,
-0.018433205783367157,
0.014740969985723495,
-0.008594237267971039,
0.011796971783041954,
0.04092227667570114,
-0.009104716591536999,
0.023356186226010323,
0.019006621092557907,
0.021985584869980812,
0.0017289872048422694,
0.018754877150058746,
0.003936985973268747,
-0.00766418594866991,
0.014461255632340908,
0.008440393954515457,
-0.0165451318025589,
0.01152424979954958,
-0.03378954902291298,
-0.03714612498879433,
1.4736927369085606e-05,
-0.02364988625049591,
-6.359141116263345e-05,

-0.012915831059217453,
-0.0099928118288517,
-0.014936771243810654,
0.00992987584322691,
-0.005943939555436373,
-0.011824943125247955,
0.014964742586016655,
-0.029202226549386978,
-0.017803847789764404,
-0.024586930871009827,
0.02613934874534607,
0.0021415664814412594,
-0.02275479957461357,
0.015692001208662987,
0.0022954095620661974,
-0.0019737377297133207,
-0.01272003073245287,
0.0005476291407831013,
-0.011657115072011948,
-0.023957572877407074,
-0.001984226983040571,
-0.015356343239545822,
-0.01966395042836666,
-0.0050313700921833515,
0.004569840617477894,
-0.014838870614767075,
-0.02145412564277649,
0.00823760125786066,
-0.02709037996828556,
-0.002333870390430093,
0.008461372926831245,
0.037118155509233475,
-0.014880827628076077,
0.01153823547065258,
0.008741088211536407,
0.0073215351440012455,
0.024782732129096985,
0.013685046695172787,
0.020503096282482147,
-0.025873620063066483,
-0.011685086414217949,
0.009824982844293118,
-0.022684870287775993,
0.007265591993927956,
-0.00487752677872777,
-0.0019073053263127804,
-0.024684831500053406,
0.003045395016670227,

-0.010321476496756077,
0.030992399901151657,
-0.0007189544849097729,
0.014894813299179077,
0.029593827202916145,
-0.015552143566310406,
-0.009531281888484955,
-0.01602765917778015,
-0.0055768140591681,
-0.032978374511003494,
-0.008825002238154411,
-0.009090730920433998,
-0.006426447536796331,
-0.007510342635214329,
-0.004929973278194666,
-0.017887761816382408,
0.024181345477700233,
-0.013971754349768162,
-0.0002190735103795305,
-0.007412442471832037,
-0.03516015037894249,
0.01711854711174965,
-0.03311823308467865,
-0.01784580573439598,
0.011796971783041954,
-0.007573278620839119,
0.011796971783041954,
-0.006237640045583248,
0.006297079846262932,
0.031887486577034,
-0.002863580361008644,
-0.004150268621742725,
0.003929993137717247,
0.030544856563210487,
0.0011573199881240726,
-0.005129270255565643,
0.0039474754594266415,
-0.002050659153610468,
0.03974747285246849,
-0.01506264228373766,
0.03560769557952881,
-0.01651715859770775,
0.0056607285514473915,
0.01829334907233715,
0.005276120733469725,
-0.020153451710939407,
0.023733802139759064,
0.012622131034731865,

0.04411102458834648,
0.009873933158814907,
0.003821603488177061,
0.009643168188631535,
0.00046677410136908293,
0.0018443695735186338,
-0.0017849301220849156,
-0.002487713471055031,
0.01848914846777916,
-0.007230627816170454,
-0.02190166898071766,
0.008838987909257412,
0.002608340699225664,
0.01781783252954483,
0.012517238035798073,
-0.019132493063807487,
0.006195683032274246,
0.014740969985723495,
0.03281054645776749,
0.0016372057143598795,
-0.023118428885936737,
-0.020852738991379738,
0.027202265337109566,
0.0037306961603462696,
-0.003540140576660633,
-0.015356343239545822,
-0.00878304522484541,
-0.025230275467038155,
0.026698779314756393,
-0.0076082427985966206,
0.009783025830984116,
0.013901825994253159,
0.02534216083586216,
-0.021314268931746483,
-0.04777529090642929,
-0.026083406060934067,
0.021468112245202065,
0.01411860529333353,
-0.01829334907233715,
-0.014251469634473324,
-0.02237718552350998,
0.0376775860786438,
-0.02310444414615631,
0.014559156261384487,
0.003720206907019019,
-0.005534856580197811,
-0.0026887585408985615,
-0.019398221746087074,

-0.013265474699437618,
-0.03406926244497299,
0.01587381586432457,
-0.013328410685062408,
0.02718827873468399,
0.03342591971158981,
-0.0050523485988378525,
0.010139661841094494,
0.014810899272561073,
0.02120238170027733,
0.03404129296541214,
0.020223380997776985,
0.01664303056895733,
-0.017761889845132828,
-0.0054054888896644115,
-0.010796991176903248,
0.014174547977745533,
-0.02053106762468815,
0.00862220861017704,
-0.00943338219076395,
-0.0010672868229448795,
0.004034886136651039,
0.0028408535290509462,
-0.01165012177079916,
0.014377341605722904,
-0.004964937921613455,
-0.013111631385982037,
-0.025985505431890488,
-0.00753132114186883,
0.00022256994270719588,
0.03247489035129547,
0.034404922276735306,
0.00828655157238245,
0.002706240862607956,
-0.009608203545212746,
0.010755034163594246,
-0.02756589464843273,
-0.008894930593669415,
0.0038950287271291018,
-0.031411971896886826,
-0.019621994346380234,
0.0051922062411904335,
-0.009824982844293118,
-0.016181502491235733,
0.044894225895404816,
-0.005394999403506517,
0.014034690335392952,
-0.008335500955581665,

-0.010559233836829662,
-0.004646762274205685,
0.01903459243476391,
-0.014768942259252071,
-0.004737669602036476,
-0.0019912198185920715,
0.029062367975711823,
-0.0019615001510828733,
-0.011964800767600536,
0.02173384092748165,
0.023342201486229897,
-0.02948193997144699,
0.006643226835876703,
0.01388084702193737,
0.008083757944405079,
0.01216060109436512,
-0.03566363826394081,
0.044166967272758484,
0.010803984478116035,
-0.01518851425498724,
0.009769040159881115,
0.005559331737458706,
0.01724441722035408,
0.005143255926668644,
-0.0024527492932975292,
-0.014475241303443909,
0.009104716591536999,
-0.009748061187565327,
-0.01532837189733982,
0.03840484470129013,
-0.0003780520346481353,
-0.0170905739068985,
0.04181736335158348,
0.008818008936941624,
-0.03261474519968033,
-0.014188533648848534,
0.021398182958364487,
0.010650141164660454,
-0.033929407596588135,
0.009810997173190117,
-0.04324391111731529,
-0.025635862722992897,
-0.021887684240937233,
-0.004856548272073269,
-0.004083835985511541,
-0.009300517849624157,
-0.0046152942813932896,
-0.012447308748960495,

-0.005905478727072477,
0.017258403822779655,
-0.025579920038580894,
0.006496376357972622,
0.018656978383660316,
-0.025454048067331314,
0.012356401421129704,
0.009076745249330997,
0.009699110873043537,
0.001924787531606853,
0.03471260890364647,
-0.007643206976354122,
0.005415977910161018,
0.017999647185206413,
-0.01864299178123474,
-0.022321240976452827,
-0.0008540042908862233,
-0.006024357862770557,
-0.002783162286505103,
-0.004335579462349415,
0.004143275786191225,
-0.0061502293683588505,
-0.008852973580360413,
0.02990151382982731,
0.0016441986663267016,
-0.0008081135456450284,
-0.017887761816382408,
-0.023593943566083908,
0.011685086414217949,
0.003982439637184143,
0.003961461130529642,
0.01903459243476391,
-0.002758687362074852,
0.014244476333260536,
-0.009475339204072952,
-0.038237012922763824,
-0.034404922276735306,
-0.019719893112778664,
-0.025006504729390144,
-0.013195546343922615,
-0.004485926125198603,
0.03295040503144264,
-0.01724441722035408,
0.02853091061115265,
0.0069474163465201855,
-0.02364988625049591,
0.0036083210725337267,
-0.006818048655986786,

0.00892290286719799,
0.005412481725215912,
0.03706221282482147,
0.030237169936299324,
0.008314522914588451,
-0.02234921231865883,
0.0008037430234253407,
-0.034125205129384995,
0.014796913601458073,
0.013754975982010365,
0.004908994771540165,
0.02467084676027298,
0.002080378821119666,
-0.032586775720119476,
-0.01669897325336933,
0.01055224146693945,
-0.008454380556941032,
0.03169168904423714,
0.00812571495771408,
-0.014671041630208492,
-0.028167281299829483,
-0.03471260890364647,
-0.008181658573448658,
0.0021433148067444563,
-0.007454399485141039,
-0.012300458736717701,
-0.029174255207180977,
0.0005179094732739031,
-0.04089430347084999,
-0.01973387971520424,
0.005835550371557474,
-0.04276839271187782,
-0.005992889869958162,
0.008342494256794453,
0.030209198594093323,
0.0233981441706419,
-0.01727238856256008,
0.020055551081895828,
-0.0025296707171946764,
0.002389813307672739,
-0.002258697059005499,
-0.006636234000325203,
-0.015985701233148575,
-0.014195526950061321,
0.0268386360257864,
0.02422330155968666,
0.025132374837994576,
-0.009950853884220123,

0.035327982157468796,
-0.007272584829479456,
0.010734056122601032,
-0.009272545576095581,
-0.0026887585408985615,
-0.023090457543730736,
0.0017613292438909411,
0.005884500220417976,
0.024335188791155815,
0.014167554676532745,
0.014097626321017742,
-0.009342474862933159,
0.007880964316427708,
0.014545169658958912,
-0.01451719831675291,
0.007356499321758747,
-0.014083640649914742,
-0.03731395676732063,
-0.015496199950575829,
-0.00406285747885704,
-0.035523779690265656,
0.00881101656705141,
-0.024880632758140564,
-0.019901707768440247,
0.048754289746284485,
-0.0077690789476037025,
-0.0050523485988378525,
0.009601211175322533,
0.011950815096497536,
-0.006415958516299725,
0.03957964479923248,
0.0019422697369009256,
0.0033740599174052477,
-0.009908896870911121,
0.008741088211536407,
-0.015985701233148575,
-0.009915890172123909,
-0.014265455305576324,
0.0013067926047369838,
0.004975426942110062,
-0.009279538877308369,
0.01893669180572033,
-0.02412540279328823,
0.04816688969731331,
-0.009020802564918995,
-0.004129289649426937,
-0.013069674372673035,
-0.04218099266290665,

-0.014824884943664074,
-0.012531223706901073,
0.016055630519986153,
-0.03289446234703064,
0.025286218151450157,
0.0049054985865950584,
-0.005664224736392498,
0.02192964032292366,
-0.04142576456069946,
0.002683514030650258,
0.0043495651334524155,
0.0019108017440885305,
0.01899263635277748,
0.05090809613466263,
0.0037796462420374155,
0.03616712614893913,
0.010943842120468616,
-0.016293387860059738,
0.01609758660197258,
-0.010587205179035664,
0.0036572711542248726,
0.03062877245247364,
0.003587342333048582,
-0.0041817366145551205,
-0.005940443370491266,
0.0004291874065529555,
-0.0029789626132696867,
-0.01899263635277748,
-0.041481707245111465,
0.013650082983076572,
-0.03085254319012165,
0.012775974348187447,
-0.03146791458129883,
0.0006096908473409712,
-0.017286375164985657,
0.01565004326403141,
-0.019230393692851067,
-0.01044035516679287,
-0.005174723919481039,
-0.01044035516679287,
-0.003919503651559353,
0.029286140576004982,
-0.009769040159881115,
0.05026474967598915,
0.01211165077984333,
-0.017677975818514824,
-0.009915890172123909,
-0.0067166518419981,

0.003587342333048582,
0.021691882982850075,
0.008587244898080826,
0.0422089658677578,
-0.027230236679315567,
0.011796971783041954,
0.021062524989247322,
0.001182669191621244,
-0.0034632189199328423,
-0.03792932629585266,
0.013650082983076572,
0.031020373106002808,
-0.03244691714644432,
-0.005943939555436373,
0.006587283685803413,
-0.02875468321144581,
0.015034670941531658,
-0.014740969985723495,
-0.0045104012824594975,
-0.036922354251146317,
-0.020600995048880577,
-0.0027866587042808533,
0.014824884943664074,
0.03725801408290863,
0.014922784641385078,
-0.0211044829338789,
-0.01874089241027832,
-0.0055733174085617065,
-0.014992713928222656,
-0.008755073882639408,
0.0027919034473598003,
-0.02766379527747631,
-0.0525863841176033,
-0.008964859880506992,
-0.03541189432144165,
0.024391131475567818,
-0.006849516183137894,
-0.003664263989776373,
-0.01606961525976658,
0.05639050528407097,
-0.027775680646300316,
0.035551752895116806,
-0.012482273392379284,
-0.026656821370124817,
-0.003381052752956748,
0.010125676169991493,
0.0029020411893725395,
-0.019552065059542656,

-0.01647520251572132,
-0.007104756310582161,
-0.042125049978494644,
0.0018915714463219047,
0.009601211175322533,
0.0062761008739471436,
0.01394378300756216,
-0.02167789824306965,
0.010796991176903248,
0.0071397204883396626,
0.011873893439769745,
-0.0037866393104195595,
-0.02253102697432041,
0.06310366094112396,
-0.022894656285643578,
-0.006115265190601349,
-0.001405566930770874,
-0.02903439663350582,
-0.0025401602033525705,
0.0009055766859091818,
0.00011942510900553316,
0.004594315774738789,
-0.01232143770903349,
-0.02023736573755741,
-0.01223752275109291,
-0.012300458736717701,
-0.016740931198000908,
-0.03219517320394516,
-0.02013946697115898,
-0.034376949071884155,
-0.03250285983085632,
0.012957788072526455,
0.003909014631062746,
0.021845726296305656,
0.012013751082122326,
0.008041800931096077,
0.01062916312366724,
0.01941220834851265,
0.0012805693550035357,
-0.017482176423072815,
-0.02990151382982731,
0.015384314581751823,
-0.0009667642880231142,
-0.040055159479379654,
-0.018475163727998734,
0.03806918486952782,
0.02374778687953949,
0.000383296690415591,

0.0037901357281953096,
-0.02282472886145115,
0.0026415567845106125,
-0.01405566930770874,
0.010587205179035664,
-0.016740931198000908,
0.02100658230483532,
-0.013055688701570034,
-0.002681765705347061,
0.027929523959755898,
-0.01102775614708662,
0.012307452037930489,
0.013622111640870571,
-0.024335188791155815,
-0.007943900302052498,
0.009713096544146538,
0.004209707956761122,
-0.024782732129096985,
-0.02046113833785057,
0.007150209974497557,
-0.0021240843925625086,
0.032782576978206635,
0.005985897034406662,
0.01704861782491207,
-0.012069693766534328,
-0.023244300857186317,
-0.022964585572481155,
-0.008978845551609993,
0.021887684240937233,
0.0010769020300358534,
0.007992850616574287,
-0.03521609306335449,
-0.014908798970282078,
0.018880749121308327,
0.0055698212236166,
0.018125519156455994,
-0.02703443542122841,
0.011901864781975746,
0.01874089241027832,
0.03222314640879631,
-0.012999746017158031,
-0.007622228469699621,
0.003699228400364518,
-0.02546803280711174,
0.02709037996828556,
-0.003877546638250351,
-0.013188553042709827,
0.012258501723408699,

-0.0111396424472332,
-0.012300458736717701,
-0.022111454978585243,
0.03384549170732498,
-0.005020880606025457,
0.011391385458409786,
0.02731415070593357,
-0.0004252539074514061,
-0.01337036769837141,
-0.004293622449040413,
0.004863541107624769,
-0.014335383661091328,
0.0012866881443187594,
-0.00020148207840975374,
0.017873777076601982,
-0.003667760407552123,
0.0032376989256590605,
-0.001823390950448811,
-0.03510420769453049,
-0.03748178482055664,
0.004769137594848871,
-0.0006595150916837156,
-0.004443969111889601,
0.002996444934979081,
-0.011468307115137577,
0.0032289577648043633,
-0.014195526950061321,
0.014223498292267323,
-0.020195409655570984,
-0.03180357441306114,
0.014587127603590488,
-0.0014213009271770716,
-0.0005852158064953983,
0.021370211616158485,
0.18539497256278992,
3.4691191103775054e-05,
0.017230432480573654,
0.038237012922763824,
-0.011971794068813324,
0.009908896870911121,
0.005611778236925602,
-0.002059400314465165,
-0.0055768140591681,
0.033677663654088974,
0.017230432480573654,
0.006101279053837061,
0.0036223067436367273,
-0.015664029866456985,

-0.004237679298967123,
0.007783064618706703,
-0.02597152069211006,
-0.02377575822174549,
-0.028055395931005478,
-0.02033526636660099,
-0.021845726296305656,
0.0027901551220566034,
-0.028670767322182655,
-0.011789979413151741,
0.02269885689020157,
-0.02464287541806698,
-0.027006464079022408,
0.011132649146020412,
-0.011048735119402409,
0.023985544219613075,
-0.015496199950575829,
0.006332044024020433,
0.023356186226010323,
-0.005346049088984728,
-0.01714651845395565,
-0.01451719831675291,
0.02818126790225506,
0.0015192010905593634,
0.03569161146879196,
0.007999843917787075,
0.020964624360203743,
-0.023216329514980316,
0.01532837189733982,
-0.0025698798708617687,
-0.010384412482380867,
0.03339794650673866,
0.00024387634766753763,
-0.028349095955491066,
-0.016307372599840164,
0.01714651845395565,
-0.00820962991565466,
0.007999843917787075,
0.03493637964129448,
0.019132493063807487,
0.015230471268296242,
0.004290125798434019,
0.008552280254662037,
0.011405371129512787,
-0.002475476125255227,
0.01518851425498724,
-0.018810821697115898,
0.02900642529129982,

0.0017115049995481968,
0.016489187255501747,
-0.035300008952617645,
-0.006839027162641287,
-0.013901825994253159,
-0.01107670646160841,
0.0056817070581018925,
-0.02718827873468399,
0.011782986111938953,
-0.03244691714644432,
0.024489032104611397,
0.0071816775016486645,
-0.027104364708065987,
-0.011377399787306786,
0.012132629752159119,
0.005804082378745079,
0.018573062494397163,
0.0319434329867363,
-0.0008448261069133878,
-0.030740657821297646,
-0.010041761212050915,
-0.03149588778614998,
-0.006279597524553537,
-0.056558333337306976,
0.011817950755357742,
0.02967774122953415,
0.002807637443765998,
-0.021691882982850075,
-0.019370250403881073,
-0.0015017188852652907,
0.007713135797530413,
0.00690196268260479,
0.007915928959846497,
-0.0016092342557385564,
-0.000676560215651989,
0.015803886577486992,
0.007720128633081913,
-0.01641925983130932,
-0.0400271899998188,
0.042125049978494644,
0.010412383824586868,
-0.007887957617640495,
-0.019705908372998238,
-0.024880632758140564,
0.010013789869844913,
-0.00046240355004556477,
-0.02269885689020157,
-0.021076511591672897,

-0.004108311142772436,
-0.023132415488362312,
-0.009027795866131783,
0.0044894227758049965,
-0.014852856285870075,
0.014083640649914742,
0.024181345477700233,
-0.015622071921825409,
0.01101377047598362,
0.01978982239961624,
0.026223262771964073,
-0.003276159754022956,
0.006762105505913496,
0.00677259499207139,
0.009573238901793957,
0.016335343942046165,
-0.030209198594093323,
-0.006426447536796331,
0.0006761231343261898,
-0.04355159401893616,
0.014978728257119656,
-0.0024072956293821335,
-0.0018566070357337594,
-0.040558647364377975,
-0.0066187516786158085,
0.022964585572481155,
0.001133719109930098,
-0.015034670941531658,
0.009517296217381954,
-0.013796932995319366,
0.011831936426460743,
0.006045336369425058,
-0.005027873907238245,
-0.015258442610502243,
-0.009251567535102367,
-0.022097470238804817,
0.013999726623296738,
0.005255141761153936,
-0.017733918502926826,
-0.015859829261898994,
-0.021146439015865326,
-0.004730676766484976,
0.009636174887418747,
-0.012454302050173283,
0.025509990751743317,
-0.029062367975711823,
-0.027523936703801155,
-0.04973329231142998,

-0.02330024354159832,
0.013272468000650406,
-0.04321593791246414,
0.010838949121534824,
0.01749616116285324,
-0.009664146229624748,
-0.038013242185115814,
-0.005811075214296579,
-0.17946502566337585,
0.007188670802861452,
0.02269885689020157,
-0.013090653344988823,
0.006975388154387474,
0.010139661841094494,
0.0024037992116063833,
-0.011104677803814411,
-0.011517257429659367,
-0.011349428445100784,
0.02948193997144699,
-0.019677937030792236,
-0.02942599728703499,
0.0014588875928893685,
-0.0018391248304396868,
0.030293114483356476,
0.003512169001623988,
-0.009776032529771328,
0.03860064223408699,
0.009209610521793365,
0.018601035699248314,
-0.025873620063066483,
0.020642952993512154,
0.00411880062893033,
0.00411880062893033,
-0.002157300477847457,
-0.007006856147199869,
0.0170905739068985,
0.011755014769732952,
0.004751655273139477,
-0.007859986275434494,
0.01520249992609024,
0.01337036769837141,
-0.011810957454144955,
0.015524172224104404,
0.012377380393445492,
-0.0067446231842041016,
-0.030013399198651314,
0.008489344269037247,
0.024726789444684982,

-0.014964742586016655,
0.02390163019299507,
0.01088789850473404,
-0.004101318307220936,
-0.013237503357231617,
0.025454048067331314,
0.02990151382982731,
-0.0034195135813206434,
0.016559116542339325,
-0.01217458676546812,
0.005982400383800268,
-0.03695032745599747,
-0.007045316975563765,
0.00500339874997735,
-0.003204482840374112,
0.022517042234539986,
-0.012251508422195911,
0.0009728830773383379,
0.0067026661708951,
0.01058021280914545,
-0.0025051957927644253,
-0.013412324711680412,
-0.012559195049107075,
-0.0075033497996628284,
-0.018629007041454315,
0.01298576034605503,
-0.0014571393840014935,
0.019020607694983482,
-0.019104521721601486,
-0.0006297953659668565,
-0.007003359496593475,
-0.016013672575354576,
0.012727024033665657,
0.013286453671753407,
0.035775523632764816,
-0.008692137897014618,
-0.01848914846777916,
0.03583146631717682,
-0.015440257266163826,
-0.0006289212615229189,
-0.019831780344247818,
0.026600878685712814,
-0.022712841629981995,
0.008706123568117619,
-0.002494706539437175,
0.004132786300033331,
-0.012391366064548492,
-0.012587166391313076,

0.015622071921825409,
-0.008594237267971039,
0.03157980367541313,
-0.016167515888810158,
0.003141546854749322,
-0.012706045061349869,
0.028726710006594658,
0.0006673820316791534,
0.00804879330098629,
0.012244516052305698,
0.01346826832741499,
0.010908877477049828,
0.005545346066355705,
0.0036048246547579765,
-0.026922550052404404,
0.012202558107674122,
0.02728617936372757,
0.004653755109757185,
-0.021845726296305656,
0.01577591523528099,
0.020405195653438568,
-0.006856509484350681,
0.0005358287016861141,
0.01567801460623741,
0.005069830920547247,
0.02065693773329258,
0.01037741918116808,
-0.010908877477049828,
0.01759406179189682,
-0.0122864730656147,
-0.00024016138922888786,
0.001053301035426557,
0.042880281805992126,
-0.02055903896689415,
-0.026377106085419655,
0.007887957617640495,
4.4415653974283487e-05,
-0.006265611853450537,
-0.07115944474935532,
-0.023244300857186317,
0.0004329023649916053,
0.015468228608369827,
-0.017831819131970406,
0.004031389486044645,
0.009741067886352539,
0.033006347715854645,
0.002608340699225664,
0.0004960567457601428,

-0.0077900574542582035,
-0.009020802564918995,
-0.0039999219588935375,
-0.004594315774738789,
0.01172704342752695,
-0.011146634817123413,
-0.008391444571316242,
0.0188387930393219,
-0.03012528456747532,
0.023062486201524734,
0.00867115892469883,
-0.024027502164244652,
0.016111573204398155,
-0.0042341831140220165,
0.0042761401273310184,
0.022559000179171562,
-0.032782576978206635,
0.001703638001345098,
0.014363355003297329,
-0.014020704664289951,
0.01714651845395565,
-0.035076238214969635,
-0.0022517042234539986,
0.0013706025201827288,
-0.019146477803587914,
-0.00811172928661108,
0.0030086825136095285,
-0.013251489028334618,
0.025579920038580894,
-0.02384568750858307,
0.0036537747364491224,
0.030516885221004486,
0.014146576635539532,
-0.05180318281054497,
-0.02071288228034973,
-0.016615059226751328,
0.014978728257119656,
0.017552103847265244,
-0.007797050289809704,
-0.019384237006306648,
-0.03960761800408363,
0.01283191703259945,
-0.022838713601231575,
-0.00045366244739852846,
0.028922511264681816,
-0.02282472886145115,
0.007866978645324707,
0.0245729461312294,

-0.0165451318025589,
-0.03742584213614464,
-0.017160503193736076,
-0.025188319385051727,
-0.010482312180101871,
0.03250285983085632,
0.02476874552667141,
-0.003174763172864914,
-0.02013946697115898,
-0.025188319385051727,
0.02744002267718315,
-0.007992850616574287,
0.008398436941206455,
0.008118722587823868,
-0.008461372926831245,
-0.006681687664240599,
-0.0353839248418808,
-0.022111454978585243,
-0.006160718854516745,
-0.01587381586432457,
-0.0005480662221089005,
-0.0015113340923562646,
-0.012741009704768658,
0.003964957315474749,
-0.005227170418947935,
0.006059322040528059,
0.014545169658958912,
0.02584564872086048,
0.012810938060283661,
0.019468151032924652,
0.047607459127902985,
-0.03681046888232231,
0.0061082723550498486,
0.0064579155296087265,
0.028055395931005478,
-0.023733802139759064,
-0.001285813981667161,
0.03202734515070915,
0.03261474519968033,
-0.018125519156455994,
-0.0009816241217777133,
0.006636234000325203,
-0.05756530910730362,
-0.01966395042836666,
-0.07362093776464462,
0.017482176423072815,
-0.014880827628076077,
-0.03720207139849663,

-0.022922629490494728,
-0.03725801408290863,
0.006730637513101101,
-0.029286140576004982,
-0.013664068654179573,
0.027328137308359146,
0.016754917800426483,
-0.0005030495813116431,
0.000714146881364286,
-0.006517354864627123,
-0.017412247136235237,
-0.008496337570250034,
0.021076511591672897,
-0.007279578130692244,
0.016754917800426483,
-7.533725147368386e-05,
-0.007950893603265285,
-0.00014717807061970234,
0.007552299648523331,
0.007915928959846497,
0.010936848819255829,
0.025300204753875732,
-0.011293485760688782,
0.007237620651721954,
-0.0019090535352006555,
-0.033929407596588135,
0.011741029098629951,
-0.037817440927028656,
-0.031887486577034,
0.018698934465646744,
0.007496356964111328,
-0.008132708258926868,
-0.0008758570183999836,
0.02297857217490673,
0.015104599297046661,
0.008076764643192291,
0.011300478130578995,
0.005108291748911142,
0.0013863365165889263,
-0.01388084702193737,
0.014475241303443909,
0.0010865172371268272,
-0.003615313908085227,
-0.0014116857200860977,
-0.005961421877145767,
-0.006346029695123434,
0.025384118780493736,
0.021538039669394493,

-0.022768786177039146,
-0.014111611992120743,
-0.005216681398451328,
-0.014922784641385078,
0.016111573204398155,
0.02100658230483532,
0.0018548588268458843,
0.006367008201777935,
0.03339794650673866,
0.0034195135813206434,
0.0032901454251259565,
-0.016083601862192154,
0.029761655256152153,
0.001652939710766077,
0.0046712374314665794,
0.005727160722017288,
0.016335343942046165,
-0.032754603773355484,
-0.006258619017899036,
-0.004335579462349415,
0.0036712568253278732,
0.02062896639108658,
0.03202734515070915,
0.002700996119529009,
-0.005262135062366724,
-0.017104560509324074,
0.006755112670361996,
0.028950482606887817,
0.003594335401430726,
-0.011363414116203785,
-0.01749616116285324,
0.0026590388733893633,
0.02636312134563923,
0.0015410537598654628,
-0.008580251596868038,
-0.004506904631853104,
0.011356420814990997,
0.0029230196960270405,
-0.03862861543893814,
0.022614942863583565,
0.006167711690068245,
0.02531418949365616,
0.0014921036781743169,
0.020978610962629318,
-0.013747982680797577,
0.01960800774395466,
0.005860025063157082,
-0.0007054057787172496,

0.026768706738948822,
0.0002628882066346705,
0.008915909565985203,
-0.03454477712512016,
-0.0423208512365818,
0.004286629613488913,
-0.01599968783557415,
-0.0002290164993610233,
0.009964840486645699,
0.023062486201524734,
0.01973387971520424,
-0.005492899566888809,
-0.0032079792581498623,
0.005436956416815519,
-0.02123035490512848,
0.010125676169991493,
-0.011992772109806538,
-0.01507662795484066,
-0.022642914205789566,
0.03994327411055565,
0.02285270020365715,
0.023006543517112732,
0.017202461138367653,
-0.014671041630208492,
-0.0068739913403987885,
0.0233981441706419,
-0.009342474862933159,
-0.027272194623947144,
0.026447035372257233,
0.0011127404868602753,
-0.002814630279317498,
-0.022195370867848396,
-0.0025051957927644253,
-0.0023723312187939882,
-0.011356420814990997,
-0.0007604746497236192,
-0.01409063395112753,
-0.009006816893815994,
0.017552103847265244,
0.09191428869962692,
0.0018566070357337594,
0.0032726633362472057,
0.01045434083789587,
-0.03317417576909065,
0.022866684943437576,
0.020153451710939407,
0.019426193088293076,
-0.005311084911227226,

```

        -0.03499232232570648,
        0.0015235715545713902,
        -0.016279401257634163,
        -0.008069772273302078,
        -0.035327982157468796,
        -0.023789744824171066,
        0.01676890254020691,
        -0.012433323077857494,
        0.013608125038444996,
        0.010936848819255829,
        0.019901707768440247,
        0.03703423961997032,
        0.00872010923922062,
        -0.0027744213584810495,
        0.0035488817375153303,
        -0.028698738664388657,
        0.00820962991565466,
        0.007776071783155203,
        0.0005096053937450051,
        0.0032551810145378113,
        0.003950971644371748,
        -0.004992909263819456,
        -0.008041800931096077,
        -0.010146655142307281,
        -0.0035803494974970818,
        -0.008153686299920082,
        0.0002735960588324815,
        -0.006374001037329435,
        0.01988772302865982,
        0.02664283476769924,
        -0.001143334317021072,
        -0.00024234666489064693,
        0.010076725855469704,
        -0.029202226549386978,
        -0.02327227219939232,
        0.009713096544146538,
        -0.020279323682188988,
        0.007412442471832037,
        -0.01781783252954483,
        -0.03569161146879196
    ],
}
},
{
    "_index": "github_issues",
    "_id": "CWLQUZMBKaR7I-arfsTE",
    "_score": 1.0,
    "_source": {

```

```

    "_type": "issue",
    "_repo": "langchain",
    "_issueNumber": "28269",
    "_title": "docs: correct HuggingFaceEmbeddings documentation model
param",
    "_createdAt": "2024-11-21T22:12:02Z",
    "_closedAt": "2024-12-31T00:36:30Z",
    "_state": "open",
    "_body": "- **Description:** Corrected the parameter name in the
HuggingFaceEmbeddings documentation under integrations/text_embedding/ from
model to model_name to align with the actual code usage in the
langchain_huggingface package.\r\n- **Issue:** Fixes #28231\r\n-
**Dependencies:** None",
    "GitHub_Issue_vector": [
        -0.01945115253329277,
        0.010854997672140598,
        0.010074162855744362,
        -0.017429348081350327,
        -0.013978336937725544,
        0.011747380718588829,
        0.011259358376264572,
        -0.015226278454065323,
        -0.01593739539384842,
        -0.023020682856440544,
        0.0021682109218090773,
        0.010081134736537933,
        0.0014126485912129283,
        0.006243192125111818,
        -0.015644581988453865,
        -0.020287761464715004,
        0.005901576951146126,
        -0.0037229086738079786,
        0.010848025791347027,
        -0.022058581933379173,
        0.013811015523970127,
        0.004378252197057009,
        -0.0164811909198761,
        0.0025185407139360905,
        -0.01735963113605976,
        0.03187479078769684,
        0.028305260464549065,
        -0.041746772825717926,
        0.00043704090057872236,
        0.003733366262167692,
        -0.007459760643541813,
        0.0009969586972147226,
        -0.0014021910028532147,
        -0.010095078498125076,

```

-0.008624041453003883,
0.0072506084106862545,
0.01836356148123741,
-0.017959199845790863,
0.0214032381772995,
0.01138485036790371,
0.01431995164602995,
0.011949560604989529,
0.006874134764075279,
-0.002682376652956009,
-0.016648512333631516,
0.014682482928037643,
-0.001096306019462645,
-0.022923078387975693,
-0.00629548029974103,
0.0128628583624959,
0.024219822138547897,
0.03725697472691536,
0.008282425813376904,
0.012967434711754322,
-0.000586933281738311,
-0.004817471839487553,
0.008651928044855595,
0.021933091804385185,
-0.012130825780332088,
0.009279384277760983,
0.0065673780627548695,
-0.002114179776981473,
-0.010241484269499779,
0.007912923581898212,
-0.027078233659267426,
0.011810125783085823,
-0.004263218492269516,
0.019297773018479347,
-0.0018370532197877765,
0.014110799878835678,
0.03279506042599678,
0.0049987370148301125,
0.00787806510925293,
-0.004671065136790276,
0.01666245609521866,
-0.014989239163696766,
0.002063634805381298,
-0.019939173012971878,
0.004587404429912567,
-0.004726839251816273,
0.005486758425831795,
0.0025586283300071955,

-0.0034370673820376396,
0.010213597677648067,
0.015491204336285591,
-0.008505521342158318,
-0.0064453729428350925,
-0.0006017482373863459,
0.017331743612885475,
-0.005856260657310486,
0.035277001559734344,
0.023522647097706795,
0.007871093228459358,
-0.007843206636607647,
0.02678542211651802,
0.02728738635778427,
-0.013999251648783684,
0.024819390848279,
0.024428972974419594,
-0.01384587399661541,
-6.465198384830728e-05,
0.036504026502370834,
-0.04473067820072174,
-0.0018405391601845622,
-0.02085944265127182,
0.0017228910000994802,
0.010931686498224735,
-0.011092036962509155,
-0.0006287636933848262,
0.011621888726949692,
-0.011280274018645287,
0.03552798181772232,
-0.006229248829185963,
-0.04478645324707031,
0.01514261681586504,
-0.00638611288741231,
0.005343838129192591,
-0.01049943920224905,
0.002244899980723858,
0.005758656654506922,
0.01231906283646822,
0.03524911403656006,
-0.009593113325536251,
-0.021347464993596077,
0.022964907810091972,
-0.018391447141766548,
-0.015588808804750443,
-0.025837264955043793,
-0.015630638226866722,
-0.024345312267541885,

0.018335673958063126,
0.015045013278722763,
0.015268108807504177,
0.008045386523008347,
-0.014640651643276215,
0.03583473712205887,
-0.01786159537732601,
0.0014997953549027443,
-0.009899869561195374,
-0.010604015551507473,
0.005120742134749889,
0.04328055679798126,
-0.0006440143915824592,
0.006842761766165495,
-0.012326034717261791,
0.005472815129905939,
-0.011573086492717266,
0.012848915532231331,
0.026590213179588318,
-0.012040193192660809,
0.0239130649715662,
-0.02076183818280697,
-0.020301703363656998,
0.006393084768205881,
0.009091148152947426,
0.023564478382468224,
-0.010262399911880493,
-0.01625809632241726,
-0.01268856506794691,
-0.0014004480326548219,
-0.00038911018054932356,
0.01776399090886116,
-0.0013185301795601845,
-0.0014265921199694276,
0.012444553896784782,
0.025711772963404655,
0.008136019110679626,
-0.023215891793370247,
-0.0019015418365597725,
-0.015825847163796425,
0.0013568747090175748,
0.007271524053066969,
-0.03229309618473053,
0.011391821317374706,
-0.00025621140957809985,
0.024401087313890457,
0.00796869769692421,
-0.011370906606316566,

-0.0021420668344944715,
-0.01478008646517992,
-0.025474734604358673,
0.007020541466772556,
0.007871093228459358,
0.04467490315437317,
-0.005253205541521311,
0.011621888726949692,
-0.009446706622838974,
-0.0009812723146751523,
0.016244152560830116,
-0.000671901332680136,
0.007466732524335384,
0.02467995695769787,
-0.00985803920775652,
-0.002703291829675436,
-0.5929882526397705,
-0.03028523363173008,
0.008909882977604866,
-0.004552545491605997,
0.013873760588467121,
0.0013237589737400413,
-0.0038588575553148985,
0.0008074145298451185,
-0.008073274046182632,
0.03661557286977768,
-0.04208141565322876,
-0.00412378367036581,
-0.008958684280514717,
0.00088410364696756,
-0.01780582219362259,
-0.01762455701828003,
0.010255428031086922,
-0.026353172957897186,
-0.00443054037168622,
0.029169756919145584,
-0.006232734769582748,
0.013692495413124561,
-0.016774004325270653,
0.0016618883237242699,
0.02176576852798462,
-0.01070161908864975,
0.004312020726501942,
-0.005044053308665752,
0.03170746937394142,
-0.003674106439575553,
-0.03176324442028999,
0.02244899980723858,

0.03106606937944889,
-0.005988724064081907,
0.03985046222805977,
-0.0007459760527126491,
-0.02686908282339573,
0.020789725705981255,
0.0092445258051157,
-0.0028252971824258566,
-0.029086096212267876,
-0.007320325821638107,
0.005399611778557301,
-0.0003122031921520829,
-0.004461912903934717,
0.01784765161573887,
0.024150103330612183,
0.005981752183288336,
-0.005821402184665203,
-0.0022989308927208185,
0.013546088710427284,
0.0011058921227231622,
0.00036601629108190536,
0.012054136954247952,
-0.009851067326962948,
-0.037145424634218216,
0.023885177448391914,
-0.02897454798221588,
0.04040820151567459,
-0.0007873707800172269,
0.01022754143923521,
0.013908619061112404,
-0.014403613284230232,
-0.0016218008240684867,
0.0024941395968198776,
0.011231471784412861,
0.026855139061808586,
0.014250234700739384,
0.005312465131282806,
-0.049666669219732285,
-0.0038065696135163307,
0.001803937484510243,
-0.03307393193244934,
0.011719493195414543,
0.003977376967668533,
0.02272786945104599,
0.009460650384426117,
0.02646472118794918,
-0.02341110073029995,
0.020231986418366432,

0.013406653888523579,
-0.021960977464914322,
-0.01006719097495079,
0.0001751649397192523,
0.03946004435420036,
0.015965282917022705,
-0.011182669550180435,
0.004618776962161064,
-0.02419193461537361,
-0.02555839531123638,
0.0022344423923641443,
0.0160768311470747,
-0.01935354806482792,
-0.030619878321886063,
0.007051913999021053,
0.029309190809726715,
-0.003750795731320977,
0.012667650356888771,
-0.017875539138913155,
-0.013922562822699547,
0.008261511102318764,
0.01671823114156723,
0.018795808777213097,
0.027259500697255135,
0.029978478327393532,
0.005992209538817406,
-0.03173535689711571,
-0.006215305533260107,
0.012988350354135036,
-0.019018903374671936,
-0.0028200685046613216,
-0.009732548147439957,
-0.019981004297733307,
0.007682856637984514,
-0.03608572110533714,
-0.03971102461218834,
0.0028235542122274637,
-0.004922047723084688,
0.014347839169204235,
0.0004762569151353091,
0.0071739195846021175,
0.002678890712559223,
-0.025335300713777542,
0.015700357034802437,
0.014187488704919815,
0.005211375188082457,
0.00024205006775446236,
-0.017011042684316635,

-0.014975295402109623,
0.014473330229520798,
0.03971102461218834,
-0.01611866056919098,
0.02546079084277153,
-0.012737367302179337,
0.006609208416193724,
0.02158450335264206,
0.04827232286334038,
-0.0044131106697022915,
0.028667790815234184,
0.026353172957897186,
-0.017819765955209732,
0.013490315526723862,
0.01821018196642399,
-0.02723161317408085,
-0.019339604303240776,
-0.04127269610762596,
-0.03990623354911804,
-0.001274085370823741,
0.003942518495023251,
-0.010283315554261208,
0.000719396339263767,
-0.027301330119371414,
-0.00605844147503376,
0.0016392301768064499,
0.013037151657044888,
-0.015323882922530174,
0.00877044815570116,
-0.026938799768686295,
0.0004923790693283081,
-0.03508179262280464,
0.014173545874655247,
0.045344192534685135,
-0.033548008650541306,
-0.010206625796854496,
-0.016634568572044373,
-0.010415778495371342,
-0.005403097718954086,
0.02268603816628456,
-0.018837638199329376,
-0.032404642552137375,
0.020524799823760986,
-0.02536318637430668,
-0.003646219614893198,
-0.0010344317415729165,
0.00634079659357667,
-0.003344691824167967,

-0.01722019538283348,
-0.002548170741647482,
-0.03151226043701172,
-0.01676006056368351,
0.005999181419610977,
0.013469399884343147,
-0.009662830270826817,
0.005880661774426699,
0.027120064944028854,
-0.0030518788844347,
0.02759414352476597,
0.014738256111741066,
-0.00571682583540678,
-0.003018762916326523,
-0.02336926944553852,
-0.0018719119252637029,
-0.024944882839918137,
0.0033394629135727882,
-0.004820957779884338,
0.011733436957001686,
0.018112577497959137,
-0.011377878487110138,
-0.007306382525712252,
0.025265581905841827,
0.011642804369330406,
0.028444696217775345,
0.02276970073580742,
-0.008805306628346443,
-0.02845863811671734,
-0.012577016837894917,
-0.001845767954364419,
-0.031540147960186005,
0.010262399911880493,
0.005235775839537382,
0.020608460530638695,
0.015546978451311588,
-0.011196613311767578,
-0.0064105140045285225,
-1.673489714448806e-05,
-0.00625016400590539,
-0.005106798838824034,
-0.005448414012789726,
-0.023257721215486526,
0.007034484762698412,
-0.010276343673467636,
0.0026893483009189367,
0.006842761766165495,
0.01539359986782074,

0.009809236973524094,
0.005699396599084139,
0.01213779766112566,
-0.0015355255454778671,
0.03179113194346428,
-0.009286356158554554,
0.0189910177141428,
0.028402864933013916,
0.007731658406555653,
0.009035374037921429,
0.007090258412063122,
-0.008672843687236309,
0.03619726747274399,
-0.002746865153312683,
0.038149356842041016,
-0.018140465021133423,
0.027538368478417397,
0.018572712317109108,
-0.011419708840548992,
-0.009739519096910954,
0.021807599812746048,
0.009293328039348125,
0.038651321083307266,
0.025851208716630936,
-0.01152428425848484,
-0.018001031130552292,
-0.0056157358922064304,
-0.010262399911880493,
-0.008352143689990044,
0.0044932859018445015,
0.0063338251784443855,
0.0004636206431314349,
-0.011057178489863873,
0.03675500676035881,
0.020120438188314438,
0.02891877293586731,
0.015421486459672451,
-0.010729506611824036,
0.0209988784044981,
0.023327438160777092,
0.026883026584982872,
-0.021389296278357506,
-0.00877044815570116,
-0.050810035318136215,
-0.017973143607378006,
-0.015463316813111305,
-0.00024727886193431914,
-0.015100786462426186,

-0.026980631053447723,
-0.0418025478720665,
0.00995564367622137,
0.0042562466114759445,
0.012270260602235794,
-0.002044462598860264,
0.022574491798877716,
0.012653706595301628,
-0.028040334582328796,
-0.03912540152668953,
0.03639247640967369,
0.008470662869513035,
-0.0038065696135163307,
0.01940932124853134,
-0.011991390958428383,
0.011977448128163815,
-0.027343161404132843,
0.013225388713181019,
0.01836356148123741,
0.012284204363822937,
-0.002595229772850871,
0.004618776962161064,
-0.0012348692398518324,
0.011705550365149975,
0.025223752483725548,
-0.020608460530638695,
0.008484606631100178,
0.02131957747042179,
-0.01040880661457777,
0.02554445154964924,
-0.008289397694170475,
0.017024986445903778,
0.054993078112602234,
0.00221527018584311,
0.007815319113433361,
-0.028486525639891624,
-0.010478523559868336,
-0.008066302165389061,
0.005835345480591059,
0.006957795470952988,
-0.008352143689990044,
-0.018781865015625954,
0.0060619269497692585,
0.020482968538999557,
-0.008665871806442738,
-0.00988592579960823,
0.037507954984903336,
-0.022853361442685127,

-0.0029089581221342087,
-0.024080386385321617,
0.013037151657044888,
0.004594375845044851,
0.0035032988525927067,
0.042973801493644714,
0.008094188757240772,
-0.003911145497113466,
-0.011642804369330406,
0.00733426958322525,
-0.012709480710327625,
-0.019297773018479347,
-0.005253205541521311,
0.0039006879087537527,
-0.028068222105503082,
-0.027747521176934242,
-0.006494175177067518,
0.0009045831393450499,
0.03056410327553749,
0.024889107793569565,
0.010332116857171059,
-0.008526436984539032,
-0.010896828025579453,
-0.018098633736371994,
-0.0030623364727944136,
-0.02313223108649254,
0.0008967399480752647,
0.007146032527089119,
0.021152256056666374,
0.019339604303240776,
-0.006201361771672964,
0.0034231238532811403,
0.007044942118227482,
-0.03229309618473053,
-0.01798708736896515,
0.0014719084138050675,
0.020566631108522415,
0.010088106617331505,
-0.01867031678557396,
0.032544076442718506,
0.009376988746225834,
-0.00906326062977314,
-0.015728242695331573,
-0.007731658406555653,
0.01575613021850586,
0.018600599840283394,
-0.015309939160943031,
0.008840165100991726,

0.0036322760861366987,
-0.013699467293918133,
-0.009620999917387962,
7.407472730847076e-05,
-0.00018671188445296139,
-0.01341362576931715,
0.008993543684482574,
-0.0010170023888349533,
-0.041244808584451675,
-0.0397946871817112,
0.04328055679798126,
0.011977448128163815,
0.011015348136425018,
-0.025488678365945816,
-0.023606307804584503,
-0.0016950040590018034,
-0.02058057300746441,
-0.030954521149396896,
-0.0072854673489928246,
-0.017554838210344315,
-0.0011137353722006083,
0.01503106951713562,
-0.004594375845044851,
0.01359489094465971,
-0.04007355496287346,
0.005476301070302725,
-0.014096856117248535,
-0.01689949631690979,
-0.0021908690687268972,
0.005957351066172123,
0.0007150389719754457,
0.005587848834693432,
-0.0023669053334742785,
-0.03176324442028999,
0.022518716752529144,
0.02587909623980522,
-0.01393650658428669,
-0.0055564758367836475,
0.014375725761055946,
-0.018698204308748245,
0.014807973988354206,
-0.017136534675955772,
-0.0065360055305063725,
-0.01780582219362259,
-0.02809610776603222,
0.009809236973524094,
0.021779712289571762,
0.0193953774869442,

-0.02104070782661438,
-0.02226773463189602,
0.013859816826879978,
0.014947408810257912,
-0.009481565095484257,
0.03360378369688988,
-0.013002293184399605,
0.008512493222951889,
-0.006989168468862772,
-0.010171767324209213,
-0.016313869506120682,
-0.014863748103380203,
-0.015407543629407883,
-0.007794404402375221,
-0.011231471784412861,
0.024401087313890457,
-0.006372169591486454,
0.018196238204836845,
0.0037717109080404043,
-0.016160491853952408,
-0.001868426101282239,
-0.0115033695474267,
0.02741287834942341,
0.003013534238561988,
0.022839417681097984,
-0.012284204363822937,
-0.01653696596622467,
-0.01872609183192253,
0.0048418729566037655,
-0.009349102154374123,
0.021110426634550095,
0.03589051216840744,
-0.015923451632261276,
0.013783128000795841,
-0.008010528050363064,
-0.015979226678609848,
-0.018823696300387383,
0.0017176622059196234,
0.0013725610915571451,
0.02845863811671734,
0.0146685391664505,
-0.009307271800935268,
-0.001580841839313507,
-0.04216507822275162,
-0.012381808832287788,
0.004409625194966793,
-0.01991128735244274,
-0.002590001095086336,

0.0025795435067266226,
-0.0160768311470747,
-0.018516939133405685,
-0.006595265120267868,
-0.0035695303231477737,
-0.03142860159277916,
-0.012639762833714485,
0.03324125334620476,
0.032683514058589935,
0.04969455674290657,
0.004005264025181532,
-0.005772599950432777,
-0.004510715138167143,
-0.017275968566536903,
0.00444483667612076,
0.005497216247022152,
-0.0033795505296438932,
0.021849431097507477,
0.014529104344546795,
0.03357589617371559,
0.06603631377220154,
-0.01070161908864975,
0.020441139116883278,
-0.016230208799242973,
0.020343534648418427,
0.021556617692112923,
-0.02463812567293644,
-0.027622031047940254,
-0.018656373023986816,
-0.010980488732457161,
0.0064767454750835896,
0.007933839224278927,
0.029002433642745018,
-0.007417930290102959,
-0.0026109162718057632,
-0.007313354406505823,
-0.01903284713625908,
0.0024714814499020576,
-0.007989613339304924,
-0.03957159072160721,
-0.016774004325270653,
0.009502480737864971,
-0.029866930097341537,
0.00593294994905591,
-0.04102171212434769,
-0.01766638644039631,
0.0008422732353210449,
-0.002989133121445775,

-0.006752129178494215,
0.0026091733016073704,
0.015198390930891037,
-0.025293469429016113,
0.011872871778905392,
-0.000324621592881158,
0.0020601488649845123,
-0.008902911096811295,
-0.006989168468862772,
-0.02381546050310135,
-0.03471926227211952,
0.0010893342550843954,
-0.007529478054493666,
0.015505148097872734,
-0.012256316840648651,
-0.001845767954364419,
-0.031317051500082016,
0.029337078332901,
0.0010649331379681826,
-0.016090773046016693,
-0.009293328039348125,
-0.02606036141514778,
-0.017819765955209732,
-0.04718473181128502,
0.014166573993861675,
-0.015686413273215294,
0.007187862880527973,
0.012918632477521896,
-0.002061891835182905,
0.02077578194439411,
-0.002924644621089101,
-0.005824888125061989,
0.003010048298165202,
0.0015093814581632614,
0.034998130053281784,
0.0029612460639327765,
0.02942073903977871,
0.03156803548336029,
-0.015379656106233597,
-0.019702134653925896,
0.03410574793815613,
-0.010199653916060925,
0.013852845877408981,
0.03223732113838196,
0.0035730162635445595,
0.0009002258302643895,
-0.03248830512166023,
0.007905951701104641,

-0.006999625824391842,
-0.022699981927871704,
-0.015965282917022705,
0.060235824435949326,
0.014891634695231915,
-0.011398793198168278,
-0.0034405533224344254,
-0.005469329189509153,
-0.022978851571679115,
-0.0024540522135794163,
-0.025767548009753227,
0.018977073952555656,
0.016732173040509224,
-0.02696668729186058,
-0.0029194157104939222,
0.029030321165919304,
0.017791878432035446,
0.013992280699312687,
0.0010274600936099887,
-0.009390932507812977,
6.2330341279448476e-06,
0.030034251511096954,
-0.01671823114156723,
0.0030483929440379143,
0.009328186511993408,
0.02897454798221588,
-0.015839790925383568,
0.014626708813011646,
0.0080872168764472,
-0.013044123537838459,
0.004880217369645834,
0.005992209538817406,
0.0007320325821638107,
0.03669923543930054,
-0.04330844432115555,
0.0009612285066395998,
0.002793924417346716,
-0.010095078498125076,
0.012054136954247952,
-0.018014973029494286,
-0.015323882922530174,
-0.017289912328124046,
-0.020050721243023872,
-0.010366976261138916,
0.02624162659049034,
0.011691606603562832,
-0.006926422938704491,
-0.0008148219785653055,

-0.00814996287226677,
-0.01708076149225235,
-0.004193501081317663,
-0.022379282861948013,
0.024470804259181023,
-0.027900900691747665,
-0.022602377459406853,
0.009662830270826817,
-0.00614907406270504,
0.01703893020749092,
-0.0024209364783018827,
0.006361711770296097,
-0.0004041430074721575,
0.04905315488576889,
-0.005298521835356951,
0.03803780674934387,
-0.02126380428671837,
-0.012625819072127342,
-0.020329590886831284,
-0.013197502121329308,
-0.014473330229520798,
-0.015533034689724445,
0.00840094592422247,
-0.00025533995358273387,
-0.03360378369688988,
0.0004902004147879779,
0.004263218492269516,
0.03212577477097511,
0.016425417736172676,
-0.014431499876081944,
0.012012306600809097,
-0.004127269610762596,
-0.0029351022094488144,
-0.025906981900334358,
-0.023397156968712807,
0.06018005311489105,
-0.01131513249129057,
-0.004580432549118996,
0.0044793421402573586,
-0.002018318511545658,
-0.0025638570077717304,
-0.03193056583404541,
0.009105090983211994,
-0.008833193220198154,
-0.04213719069957733,
-0.004008749965578318,
0.004092410672456026,
-0.008073274046182632,

-0.033046044409275055,
-0.0049185617826879025,
-0.0343567319214344,
-0.037647392600774765,
-0.014710369519889355,
0.030006365850567818,
0.016160491853952408,
0.007801375817507505,
0.0004710281209554523,
0.020231986418366432,
0.03625304251909256,
0.021988864988088608,
-0.007459760643541813,
-0.014654595404863358,
-0.019018903374671936,
-0.016871608793735504,
-0.03424518182873726,
-0.01204716507345438,
-0.027482595294713974,
0.02759414352476597,
0.020441139116883278,
-0.02145901322364807,
-0.0005250591202639043,
-0.010213597677648067,
-0.0239130649715662,
-0.02058057300746441,
-0.008393974043428898,
0.008665871806442738,
0.003362121060490608,
0.0034405533224344254,
-0.012618848122656345,
0.024066442623734474,
-0.0025586283300071955,
0.014292065054178238,
-0.03042466938495636,
0.0015433687949553132,
0.0008923825807869434,
0.0029333592392504215,
-0.0035799879115074873,
-0.011454567313194275,
-0.025349242612719536,
0.004991765134036541,
0.015114730224013329,
0.03187479078769684,
0.011796182952821255,
0.03237675502896309,
-0.003939032554626465,
-0.020064665004611015,

-0.012702508829534054,
0.013525173999369144,
0.0026091733016073704,
-0.013183558359742165,
0.0021281233057379723,
0.0119565324857831,
-0.015784017741680145,
0.019479038193821907,
-0.016021056100726128,
-0.020301703363656998,
-0.04724050313234329,
0.008261511102318764,
0.03385476395487785,
0.03179113194346428,
-0.02787301316857338,
0.0006501146708615124,
-0.012102939188480377,
-0.012360893189907074,
0.019869456067681313,
0.005340352188795805,
-0.019562700763344765,
0.022755756974220276,
0.008317284286022186,
0.031010296195745468,
0.005134685896337032,
0.02700851671397686,
-0.006197875831276178,
-0.00849855039268732,
0.04032453894615173,
0.020566631108522415,
0.0008265867945738137,
-0.021696051582694054,
0.008268482983112335,
-0.02505643106997013,
-0.015351769514381886,
-0.00611421512439847,
0.00364970532245934,
-0.02449869178235531,
0.0009708146681077778,
-0.022964907810091972,
-0.08053752779960632,
-0.011991390958428383,
-0.01250032801181078,
0.028123995289206505,
0.02536318637430668,
-0.01395742129534483,
-0.01917228288948536,
0.008693758398294449,

-0.01268159318715334,
0.015170504339039326,
-0.004632720723748207,
-0.005141657777130604,
-0.00045664890785701573,
0.03990623354911804,
-0.009181780740618706,
0.009112062864005566,
0.16241362690925598,
-0.014863748103380203,
0.009662830270826817,
0.025516565889120102,
0.0042248740792274475,
-0.0052218325436115265,
-0.01293257623910904,
0.013455456122756004,
-0.019186226651072502,
0.015477260574698448,
0.029030321165919304,
0.002455795183777809,
-0.003449267940595746,
-0.0014762657228857279,
-0.011984419077634811,
0.010917743667960167,
-0.0418025478720665,
-0.031595923006534576,
-0.05761445313692093,
0.004977821838110685,
0.00024074286920949817,
-0.016843721270561218,
-0.01143365167081356,
-0.010555213317275047,
0.019841568544507027,
-0.030257347971200943,
-0.018377503380179405,
0.015979226678609848,
-0.0022710440680384636,
0.024610238149762154,
-0.020915217697620392,
0.021166199818253517,
0.026673873886466026,
-0.012165684252977371,
-0.03410574793815613,
-0.020008891820907593,
0.011210556142032146,
0.027705691754817963,
0.02186337299644947,
0.007703771814703941,

0.013908619061112404,
-0.038456112146377563,
0.016550907865166664,
-0.011858928017318249,
0.0038205129094421864,
0.02905820868909359,
-0.021737882867455482,
-0.012186599895358086,
-0.006002667360007763,
0.02131957747042179,
-0.018781865015625954,
0.007030998822301626,
0.04715684428811073,
0.049527235329151154,
0.028263431042432785,
-0.020245930179953575,
0.0019991460721939802,
0.032181546092033386,
0.014459386467933655,
0.006675440352410078,
-0.033408574759960175,
-0.010827111080288887,
0.0035032988525927067,
0.014905578456819057,
-0.02455446496605873,
-0.014361782930791378,
-0.023355325683951378,
-0.0022780157160013914,
-0.003583473851904273,
-0.03142860159277916,
0.005566933657974005,
-0.027761464938521385,
0.002309388481080532,
0.014891634695231915,
-0.048300206661224365,
-0.023564478382468224,
0.021305633708834648,
0.006438401062041521,
0.007320325821638107,
0.0703309029340744,
-0.0049081044271588326,
-0.017777934670448303,
-0.026659930124878883,
-0.012834971770644188,
-0.007271524053066969,
-0.019548757001757622,
0.0012671136064454913,
0.004196987021714449,

-0.014501216821372509,
-0.021486900746822357,
-0.01042972132563591,
-0.018642431125044823,
-0.0011346505489200354,
0.024944882839918137,
-3.888923311023973e-05,
0.03399420157074928,
0.004901132546365261,
0.006351254414767027,
0.005493730306625366,
0.001652302104048431,
-0.015993168577551842,
0.04364308714866638,
0.011029290966689587,
-0.032320983707904816,
-0.03538854792714119,
0.016341757029294968,
0.011545199900865555,
-0.006783502176403999,
-0.01748512126505375,
-0.002705034799873829,
0.010380919091403484,
-0.034635599702596664,
-0.012639762833714485,
0.012346950359642506,
0.007843206636607647,
0.005974780302494764,
0.021486900746822357,
-0.018698204308748245,
0.015909507870674133,
0.002834012033417821,
-0.018684260547161102,
0.005619221832603216,
0.033910539001226425,
0.021194087341427803,
-0.00553556066006422,
-0.0030083053279668093,
-0.031317051500082016,
0.0038344564381986856,
-0.009349102154374123,
0.0014832374872639775,
0.00715300440788269,
0.0057272836565971375,
0.021194087341427803,
-0.012577016837894917,
-0.0006357354577630758,
-0.012563074007630348,

-0.006807903293520212,
0.0019085136009380221,
0.010513382963836193,
-0.011984419077634811,
-0.012312090955674648,
-0.015825847163796425,
-0.011099008843302727,
-0.019060734659433365,
-0.025948813185095787,
-0.0389859639108181,
0.012667650356888771,
0.013811015523970127,
0.004782612901180983,
0.008847136981785297,
-0.02536318637430668,
-0.027663860470056534,
0.01822412572801113,
-0.007431874051690102,
0.013734325766563416,
-0.0014753942377865314,
-0.010834082961082458,
-0.042611267417669296,
-0.00733426958322525,
0.0001050662831403315,
-0.042471833527088165,
-0.0007390043465420604,
0.007431874051690102,
-0.02983904257416725,
-0.027942730113863945,
-0.021696051582694054,
-0.1770264059305191,
0.0115033695474267,
0.029588060453534126,
-0.06140707805752754,
0.004406139254570007,
-0.002703291829675436,
0.022964907810091972,
-0.004217902198433876,
-0.014250234700739384,
-0.018781865015625954,
0.029225530102849007,
0.006888078060001135,
-0.025906981900334358,
-0.019381433725357056,
-0.0033307482954114676,
0.042109303176403046,
-0.029615947976708412,
-0.008122076280415058,

0.008275453932583332,
0.006372169591486454,
0.039599478244781494,
-0.04272281751036644,
0.01484980434179306,
0.009195723570883274,
-0.02956017293035984,
-0.008875023573637009,
-0.00824756734073162,
0.03374321758747101,
0.0025655999779701233,
-0.014766143634915352,
-0.013239332474768162,
-0.011280274018645287,
0.05716826021671295,
-0.007445817347615957,
0.020984934642910957,
0.013364823535084724,
0.023857291787862778,
-0.006438401062041521,
-0.02801244705915451,
0.02122197300195694,
-0.0022536145988851786,
0.011998362839221954,
-0.010966545902192593,
-2.8213757104822434e-05,
-0.024624181911349297,
0.019102565944194794,
0.03145648539066315,
-0.0005673252744600177,
0.006326853297650814,
-0.020524799823760986,
0.028667790815234184,
0.002879328327253461,
-0.005518131423741579,
0.0010858483146876097,
-0.004186529200524092,
0.005650594364851713,
-0.0096976887434721,
0.003757767379283905,
-0.0016470733098685741,
-0.017680330201983452,
-0.000594340730458498,
-0.0031982853543013334,
0.0056471084244549274,
-0.005546018481254578,
0.010325145907700062,
-0.007146032527089119,

-0.011552171781659126,
0.012326034717261791,
-0.013678551651537418,
0.012158713303506374,
-0.014724313281476498,
-0.006832304410636425,
0.0007189605967141688,
-0.0017751790583133698,
0.021919148042798042,
0.026576269418001175,
-0.0286399032920599,
0.011468511074781418,
0.018252013251185417,
-0.011900758370757103,
-0.020566631108522415,
0.05028018355369568,
-0.022504772990942,
0.012451525777578354,
0.015240221284329891,
-0.012214486487209797,
-0.016369642689824104,
-0.007738630287349224,
0.016188377514481544,
0.018154408782720566,
0.019939173012971878,
-0.025628112256526947,
0.02724555693566799,
-0.01131513249129057,
0.0060305544175207615,
0.009927756153047085,
0.01699710078537464,
-0.014794030226767063,
0.022114356979727745,
0.015616695396602154,
-0.00018333495245315135,
-0.003670620732009411,
-0.012618848122656345,
0.037507954984903336,
0.024401087313890457,
0.007940811105072498,
-0.01880975253880024,
0.011691606603562832,
0.04517686739563942,
-0.039599478244781494,
-0.00877044815570116,
0.02678542211651802,
0.014103827998042107,
0.009648886509239674,

-6.988079258007929e-05,
0.0198973435908556,
-0.01422931905835867,
-0.021556617692112923,
0.005085883662104607,
0.017917368561029434,
0.041244808584451675,
-0.020873386412858963,
0.00634079659357667,
0.005549504421651363,
-0.022937022149562836,
-0.03251619264483452,
-0.0769679993391037,
-0.021068595349788666,
0.014131715521216393,
-0.0026736620347946882,
-0.015630638226866722,
0.015003182925283909,
0.004824443254619837,
0.02017621323466301,
-0.0006780016119591892,
-0.0019015418365597725,
-0.017513008788228035,
0.011426680721342564,
0.010757393203675747,
-0.0027451221831142902,
0.027635972946882248,
0.006807903293520212,
0.001934657571837306,
-0.008052358403801918,
-0.03299026936292648,
0.037870485335588455,
-0.01766638644039631,
-0.027259500697255135,
0.004050580319017172,
-0.026297399774193764,
0.010471551679074764,
-0.00842883251607418,
-0.041830435395240784,
-0.010081134736537933,
0.01913045160472393,
-0.0020130896009504795,
-0.0008514236542396247,
-0.04213719069957733,
-0.00602358253672719,
-0.006654525175690651,
-0.02554445154964924,
-0.010450636968016624,

-0.00629548029974103,
0.0022326994221657515,
0.0011180926812812686,
-0.02518192119896412,
0.020915217697620392,
0.015100786462426186,
0.00821968074887991,
-0.005479787010699511,
-0.01940932124853134,
0.007675884757190943,
-0.0239130649715662,
0.0311497300863266,
-0.0038518859073519707,
-0.03128916397690773,
-0.044200826436281204,
0.010471551679074764,
-0.02919764257967472,
0.014570934697985649,
0.0036601629108190536,
0.021919148042798042,
-0.01165674813091755,
0.0164811909198761,
0.005936435889452696,
-0.03767527639865875,
-0.036643460392951965,
0.005507674068212509,
-0.013106869533658028,
0.0008688530069775879,
0.009913813322782516,
-0.00831031333655119,
-0.0025411988608539104,
-0.0282355435192585,
0.004280647728592157,
-0.01088288426399231,
0.00184751080814749,
0.014361782930791378,
-0.011140839196741581,
0.0038762870244681835,
-0.03625304251909256,
-0.023285608738660812,
0.003004819620400667,
-0.029030321165919304,
0.003193056443706155,
-0.016202321276068687,
0.0025638570077717304,
0.003072794061154127,
0.007668912876397371,
-0.009279384277760983,

0.0007712486549280584,
0.024164047092199326,
0.020733952522277832,
0.0019172282190993428,
-0.008916853927075863,
-0.03457982465624809,
0.0024505662731826305,
-0.0016984898829832673,
0.02700851671397686,
-0.03271140158176422,
0.007271524053066969,
0.031623806804418564,
0.0012069822987541556,
-0.018572712317109108,
-0.006389598827809095,
0.008986571803689003,
-0.04286225140094757,
-0.008324256166815758,
-0.07300805300474167,
0.02877933904528618,
-0.014989239163696766,
-0.024526577442884445,
0.007529478054493666,
-0.028361033648252487,
0.019981004297733307,
-0.009844095446169376,
0.006529033649712801,
0.023480817675590515,
-0.007201806642115116,
0.02095704711973667,
-0.008847136981785297,
0.023160116747021675,
-0.027078233659267426,
-0.002046205336228013,
0.028723565861582756,
-0.023160116747021675,
0.03524911403656006,
0.020566631108522415,
-0.02437319979071617,
-0.0014692939585074782,
0.008631013333797455,
0.022839417681097984,
-0.007801375817507505,
-0.016774004325270653,
0.0035085277631878853,
-0.0030937092378735542,
0.014347839169204235,
-0.004542088136076927,

0.006846247706562281,
-0.02136140875518322,
-0.018740033730864525,
0.03271140158176422,
0.013727353885769844,
-0.0013603605329990387,
0.013078982941806316,
0.01006719097495079,
0.03589051216840744,
-0.008554323576390743,
-0.017345687374472618,
-0.020929161459207535,
0.029225530102849007,
-0.023243777453899384,
-0.0031756272073835135,
0.01024845615029335,
0.008979599922895432,
0.0021403238642960787,
0.011203584261238575,
0.021235916763544083,
0.047575145959854126,
0.016634568572044373,
-0.014515160582959652,
-0.01699710078537464,
0.0059050628915429115,
-0.018600599840283394,
0.03173535689711571,
0.03293449431657791,
0.013525173999369144,
0.0019398863660171628,
0.04710106924176216,
-0.02505643106997013,
0.011719493195414543,
-0.016160491853952408,
0.011935616843402386,
0.004765183664858341,
-0.01876792125403881,
-0.00015871599316596985,
0.022839417681097984,
-0.02232350781559944,
-0.004277161788195372,
-0.023425042629241943,
-0.013838902115821838,
-0.004137726966291666,
0.02035747841000557,
0.028807226568460464,
0.0035538440570235252,
-0.032265208661556244,

0.014821916818618774,
0.014501216821372509,
0.010757393203675747,
-0.020984934642910957,
-0.023341381922364235,
-0.0030152772087603807,
0.01671823114156723,
0.019144395366311073,
-0.002370391273871064,
0.003538157558068633,
0.010750421322882175,
-0.0228115301579237,
-0.007982641458511353,
0.020329590886831284,
0.0007943425443954766,
0.031902678310871124,
0.014766143634915352,
-0.006211819592863321,
0.02401066944003105,
0.03034100867807865,
0.022699981927871704,
-0.006089814007282257,
0.014710369519889355,
0.01649513468146324,
0.002901986474171281,
-0.03957159072160721,
-0.048579078167676926,
0.008784390985965729,
-0.036587685346603394,
-0.025572339072823524,
-0.006546462886035442,
0.0029786755330860615,
0.005113770719617605,
-0.021612390875816345,
-0.006354740355163813,
0.011844984255731106,
-0.01802891679108143,
0.004789584781974554,
-0.02654838189482689,
-0.015546978451311588,
-0.03181901574134827,
0.04442392289638519,
0.011810125783085823,
-0.025990642607212067,
0.07278495281934738,
0.002590001095086336,
0.026395004242658615,
0.017526952549815178,

-0.004095896612852812,
0.007146032527089119,
0.02797061763703823,
0.02527952566742897,
-0.0006544719799421728,
-0.0006806160090491176,
-0.017415404319763184,
-0.022992795333266258,
-0.013873760588467121,
-0.018293842673301697,
-0.021291691809892654,
0.010436693206429482,
0.01402713917195797,
0.06977316737174988,
6.323584966594353e-05,
-0.018684260547161102,
0.013831930235028267,
-0.034412503242492676,
0.020259873941540718,
0.013176586478948593,
0.014863748103380203,
0.0018614543369039893,
-0.013518202118575573,
-0.024387143552303314,
-0.011677662841975689,
-0.04790979251265526,
-0.028179768472909927,
-0.03753584250807762,
0.005852775182574987,
0.0023895634803920984,
0.03042466938495636,
0.019506925716996193,
0.02258843369781971,
0.03212577477097511,
0.003733366262167692,
0.004960392136126757,
0.010889856144785881,
0.00861009769141674,
0.011280274018645287,
0.021974921226501465,
-0.01213779766112566,
-0.022100413218140602,
0.002236185362562537,
-0.003712451085448265,
0.01514261681586504,
-0.022797586396336555,
-0.0020793210715055466,
0.006895049940794706,

```

        0.014752199873328209,
        -0.009467621333897114,
        -0.010206625796854496,
        0.009488536976277828,
        0.0030449070036411285,
        0.010011416859924793,
        0.0009568711975589395,
        -0.028068222105503082,
        -0.03544432297348976,
        0.024401087313890457,
        -0.0049987370148301125,
        0.013260248117148876,
        -0.02956017293035984,
        -0.03187479078769684
    ],
}
},
{
  "_index": "github_issues",
  "_id": "CmLQUZMBKaR7I-arfsTE",
  "_score": 1.0,
  "_source": {
    "_type": "issue",
    "_repo": "langchain",
    "_issueNumber": "28268",
    "_title": "standard-tests: troubleshooting docstrings",
    "_createdAt": "2024-11-21T21:41:43Z",
    "_closedAt": "2024-11-21T22:05:32Z",
    "_state": "closed",
    "_body": "None",
    "GitHub_Issue_vector": [
      -0.016303295269608498,
      0.002488586585968733,
      0.035421960055828094,
      -0.011929415166378021,
      -0.01394039485603571,
      0.01287026610225439,
      0.0039429557509720325,
      -0.029417751356959343,
      -0.03947264701128006,
      -0.013976304791867733,
      0.0254963431507349,
      -0.0022892842534929514,
      -0.009868161752820015,
      0.013250916264951229,
      -0.01183604821562767,
      0.008869854733347893,
      0.01126148272305727,

```

-0.03013595938682556,
0.021675482392311096,
0.009221775457262993,
0.003565897000953555,
0.018716469407081604,
-0.03088289313018322,
0.025654347613453865,
-0.012970815412700176,
0.02206331305205822,
0.0037562218494713306,
-0.018860111013054848,
0.009832251816987991,
0.012489616870880127,
0.026659836992621422,
-0.016533121466636658,
-0.009817887097597122,
-0.02016724832355976,
0.01091674342751503,
-0.0009053894900716841,
0.020368345081806183,
-0.009106862358748913,
0.020784905180335045,
-0.011362032033503056,
0.006180169992148876,
0.005476327147334814,
0.014881245791912079,
-0.02928847447037697,
-0.015139800496399403,
0.011886322870850563,
0.024203570559620857,
-0.009889707900583744,
-0.0001699007989373058,
0.01194377988576889,
0.025453250855207443,
-0.012970815412700176,
0.0004428194079082459,
0.012590166181325912,
0.014076854102313519,
0.004416971933096647,
0.012575801461935043,
0.03203202411532402,
-0.0011913256021216512,
-0.01532653346657753,
-0.01321500539779663,
-0.01194377988576889,
-0.02019597589969635,
0.0014418002683669329,
-0.023025711998343468,

-0.004388243891298771,
-0.016777312383055687,
0.01901811733841896,
0.011268665082752705,
-0.011225572787225246,
0.02321244403719902,
0.01381111703813076,
-0.006840920075774193,
-0.017294419929385185,
0.0304232407361269,
-0.016561849042773247,
0.005023856647312641,
-0.003352230414748192,
-0.014163038693368435,
0.001791028305888176,
0.02696148492395878,
-0.0007756633567623794,
-0.02110091596841812,
0.022120770066976547,
0.02996358834207058,
0.012374703772366047,
0.01646129973232746,
0.02528087981045246,
0.0069845616817474365,
-0.01624583825469017,
0.01832863874733448,
0.010485819540917873,
0.022005857899785042,
0.01529780589044094,
0.015613816678524017,
0.04174217954277992,
-0.005720517598092556,
0.05030320584774017,
0.03231930732727051,
-0.047258008271455765,
-0.011555948294699192,
0.006693687755614519,
-0.02322680875658989,
-0.004973582457751036,
-0.01832863874733448,
-0.025180332362651825,
0.00916431937366724,
-0.00018549936066847295,
0.03085416555404663,
0.007426258642226458,
-0.010507365688681602,
0.04185709357261658,
-0.0012828969629481435,

-0.048349685966968536,
0.02572616934776306,
-0.0033899361733347178,
-0.007548353634774685,
-0.021991493180394173,
0.00976043101400137,
0.005515828263014555,
0.0395013764500618,
0.0015504290349781513,
0.006320219952613115,
-0.015398354269564152,
-0.005066949408501387,
-0.00203252537176013,
-0.007174886297434568,
-0.026300733909010887,
-0.011397942900657654,
-0.009911254048347473,
0.03217566758394241,
0.01427076943218708,
0.006880421657115221,
-0.015383990481495857,
-0.010033349506556988,
0.016418207436800003,
-0.009006313979625702,
0.014823788776993752,
-0.01739496923983097,
-0.012949269264936447,
-0.006819373928010464,
0.03901299461722374,
-0.020267797634005547,
-0.016073469072580338,
-0.02439030446112156,
0.026315098628401756,
-0.01854410022497177,
-0.002610681811347604,
0.03269277513027191,
-0.015843642875552177,
0.005745654460042715,
-0.024921776726841927,
-0.00869030226022005,
-0.009322324767708778,
-0.01042836345732212,
0.02576926164329052,
-0.018917568027973175,
0.009868161752820015,
-0.036341264843940735,
-0.0013592063914984465,
0.00857538916170597,

0.02023906819522381,
0.006805009674280882,
-0.028254257515072823,
0.0012918745633214712,
0.037289299070835114,
0.011864776723086834,
0.0046288431622087955,
-0.004075823817402124,
-0.0008268356323242188,
-0.014809424988925457,
0.007885911501944065,
-0.035680513828992844,
0.026056544855237007,
-0.011871959082782269,
0.03223312273621559,
0.0021061415318399668,
0.0030972668901085854,
-0.00987534411251545,
0.007778179831802845,
-0.03499103710055351,
0.00823064986616373,
0.0027417545206844807,
0.03740421310067177,
-0.012123331427574158,
0.009293596260249615,
0.014313862659037113,
0.009559333324432373,
0.022580422461032867,
0.01311445701867342,
0.014967430382966995,
0.018931932747364044,
-0.028239892795681953,
0.009228957816958427,
-0.599846363067627,
-0.013839845545589924,
-0.012037146836519241,
-0.004880215507000685,
0.005325503647327423,
0.017596067860722542,
-0.007857182994484901,
0.0028638497460633516,
-0.012532709166407585,
0.015154164284467697,
-0.013796753250062466,
0.01185041293501854,
0.016375115141272545,
-0.02131637930870056,
-0.00880521535873413,

-0.007505261339247227,
0.021675482392311096,
-0.007027653977274895,
-0.016145288944244385,
0.018414823338389397,
-0.03433028608560562,
0.02092854678630829,
0.02342790737748146,
-0.007871546782553196,
0.006140668410807848,
-0.020827997475862503,
0.026573652401566505,
0.006277127657085657,
0.000917060358915478,
-0.0020917775109410286,
-0.030279600992798805,
0.010945471934974194,
0.01854410022497177,
0.0012730215676128864,
0.051682163029909134,
0.002738163573667407,
-0.013408921658992767,
0.023499727249145508,
-0.004011185374110937,
0.01877392642199993,
-0.021431291475892067,
-0.0037274935748428106,
0.01903248205780983,
9.280579251935706e-05,
-0.0069881523959338665,
-0.00893449317663908,
0.005878522992134094,
0.0021653936710208654,
-0.0033127290662378073,
0.008352745324373245,
0.018630284816026688,
-0.0038962720427662134,
-0.003249885980039835,
0.004937672056257725,
0.0248643197119236,
-0.02365773357450962,
0.01336582936346531,
-0.02782333269715309,
0.02226441167294979,
-0.011527219787240028,
-0.003201406914740801,
-0.000895514152944088,
0.02066999301314354,

0.012898994609713554,
-0.002343149855732918,
0.030250871554017067,
0.022221319377422333,
0.020784905180335045,
0.006194533780217171,
-0.05498591437935829,
0.015915462747216225,
-0.00673318887129426,
-0.024203570559620857,
-0.013674657791852951,
0.020526351407170296,
0.029992317780852318,
0.01241061370819807,
0.01929103583097458,
-0.011936597526073456,
0.008905764669179916,
-0.009494693949818611,
-0.015455811284482479,
-0.025352701544761658,
-0.007228751666843891,
0.037260569632053375,
0.0076129925437271595,
-0.003262454643845558,
0.0058821141719818115,
-0.00834556296467781,
-0.00030456457170657814,
-1.6117572158691473e-05,
0.006115531083196402,
0.004251784645020962,
-0.015671273693442345,
0.005716926418244839,
0.02643001079559326,
0.002070231130346656,
0.009731702506542206,
0.014529324136674404,
-0.020483259111642838,
-0.018716469407081604,
0.018026990815997124,
-0.006262763403356075,
0.00695942435413599,
0.011455398984253407,
-0.0004174577188678086,
-0.03812241926789284,
-0.00353896408341825,
0.012144877575337887,
0.0055625117383897305,
-0.022680971771478653,

0.007548353634774685,
0.0007011494017206132,
0.022853340953588486,
0.014127128757536411,
-0.04381061717867851,
0.004395425785332918,
-0.009501876309514046,
0.00018875373643822968,
0.009810705669224262,
-0.011814502067863941,
0.005357822868973017,
0.0010943677043542266,
0.022393688559532166,
0.01626020297408104,
-0.0025604073889553547,
-0.003680810099467635,
-0.021689847111701965,
-0.01715077832341194,
0.006553637329488993,
-0.018213724717497826,
0.01947776973247528,
0.02066999301314354,
-0.005566102918237448,
-5.456688813865185e-05,
0.023571548983454704,
0.028196800500154495,
-0.0021366653963923454,
0.026573652401566505,
0.01929103583097458,
-0.007850000634789467,
-0.008905764669179916,
0.01765352487564087,
-0.01996614970266819,
-0.005788747221231461,
-0.041943278163671494,
-0.01763916015625,
-0.0005314730806276202,
-0.011792955920100212,
-0.0009237935300916433,
-0.016662398353219032,
-0.011462581343948841,
-0.018012627959251404,
-0.0003784052387345582,
0.002574771409854293,
-0.008180376142263412,
0.022106407210230827,
-0.04197200760245323,
-0.009918436408042908,

-0.010686918161809444,
0.00895603932440281,
-0.005038220901042223,
-0.021531840786337852,
-0.0016913771396502852,
-0.015398354269564152,
-0.011290211230516434,
-0.027248768135905266,
0.019765052944421768,
-0.012209516018629074,
-0.03200329840183258,
-0.008094190619885921,
-0.02156056836247444,
-0.005684607196599245,
-0.019133029505610466,
-0.006284310016781092,
0.013444831594824791,
-0.013775207102298737,
0.00521777244284749,
-0.02154620550572872,
-0.00423742039129138,
-0.013523834757506847,
0.005512237548828125,
-0.014601144939661026,
0.0010010007536038756,
0.017596067860722542,
0.01670549064874649,
0.027952609583735466,
0.00626994576305151,
0.013681840151548386,
-0.019765052944421768,
-0.01693531684577465,
-0.001781152910552919,
-0.016992773860692978,
0.021747302263975143,
-0.017840256914496422,
0.000832222169265151,
-0.007957731373608112,
0.01486688107252121,
-0.00214743847027421,
0.04849332571029663,
0.006420769263058901,
0.01761043071746826,
0.02740677259862423,
-0.017754072323441505,
0.0017892328323796391,
-0.011534402146935463,
0.0074047124944627285,

-0.027277495712041855,
0.0018709287978708744,
-0.009932800196111202,
0.018486643210053444,
-0.01162058673799038,
-0.01692095212638378,
-0.00869030226022005,
0.007634538691490889,
-0.009106862358748913,
-0.011570312082767487,
-0.008798033930361271,
-0.00964551791548729,
-0.0018251431174576283,
-0.014191767200827599,
0.014780696481466293,
0.0032247486524283886,
0.00778536219149828,
0.006119122263044119,
-0.01394039485603571,
0.013329918496310711,
0.02019597589969635,
0.028498446568846703,
0.001391525729559362,
0.007300572469830513,
0.02667420171201229,
0.018845748156309128,
-0.005602013319730759,
0.026042180135846138,
0.015929827466607094,
0.03855334222316742,
-0.01696404628455639,
0.03226185217499733,
-0.028972463682293892,
0.014435957185924053,
0.024519581347703934,
0.008008006028831005,
-0.030279600992798805,
0.030049774795770645,
0.003077516332268715,
0.028972463682293892,
0.0075555359944701195,
0.0019086346728727221,
0.011419489048421383,
0.01116093434393406,
-0.011218390427529812,
-0.019808145239949226,
-0.014924338087439537,
0.012597347609698772,

-0.011103477329015732,
-0.03456011414527893,
0.026990212500095367,
0.004567795433104038,
0.025122875347733498,
0.017323149368166924,
-0.01451496034860611,
0.02457703836262226,
0.030250871554017067,
0.040736690163612366,
-0.009702973999083042,
-0.015671273693442345,
-0.015628181397914886,
0.002160007134079933,
-0.007447804789990187,
-0.00019627246365416795,
-0.009063770063221455,
-0.0025927266106009483,
-0.039846114814281464,
0.054411351680755615,
0.017538610845804214,
0.014924338087439537,
0.018960660323500633,
0.018716469407081604,
-0.017323149368166924,
-0.03895553946495056,
-0.020986003801226616,
0.02457703836262226,
0.012209516018629074,
0.004833532031625509,
-0.01299954392015934,
-0.02413174882531166,
0.013222187757492065,
-0.03734675422310829,
-0.0012631462886929512,
0.012446524575352669,
-0.009839433245360851,
-0.0012083830079063773,
-0.022451145574450493,
-0.008654392324388027,
-0.010536094196140766,
0.034904852509498596,
-0.0120802391320467,
0.015139800496399403,
0.028929371386766434,
0.0018745198613032699,
0.005928797647356987,
0.03456011414527893,

-0.0004910739371553063,
0.029791219159960747,
0.020023606717586517,
-0.00973888486623764,
-0.026085272431373596,
-0.018199361860752106,
-0.010615097358822823,
0.027492957189679146,
0.008762123063206673,
-0.001852973597124219,
-0.020267797634005547,
-0.010514548048377037,
0.013243733905255795,
0.0023754690773785114,
-0.008532296866178513,
0.012575801461935043,
0.006305856164544821,
-0.01834300346672535,
-0.04067923501133919,
0.005648696795105934,
0.0005615479894913733,
0.029762491583824158,
0.046309977769851685,
0.02439030446112156,
-0.001852973597124219,
-0.020483259111642838,
0.001791028305888176,
-0.008632846176624298,
-0.013847027905285358,
0.002723799319937825,
-0.009487512521445751,
-0.017696617171168327,
-0.012152059935033321,
0.009473147802054882,
0.00985379796475172,
0.025438886135816574,
0.013085728511214256,
-0.005005901679396629,
-0.008166011422872543,
-0.013021090067923069,
-0.0012389067560434341,
0.010758738033473492,
-0.016346387565135956,
-0.00047940306831151247,
0.017825894057750702,
0.03826605901122093,
0.002905146684497595,
-0.019089937210083008,

0.01056482270359993,
0.013638747856020927,
-0.02085672691464424,
-0.00046548779937438667,
0.008381473831832409,
0.0008012495236471295,
0.03599652647972107,
-0.031486187130212784,
0.04225929081439972,
0.011871959082782269,
-0.005303957499563694,
0.007627356797456741,
0.007383166346698999,
0.010866469703614712,
-0.003172678640112281,
-0.007160522043704987,
-0.028455354273319244,
0.005361414048820734,
-0.014996158890426159,
-0.028886279091238976,
0.020368345081806183,
0.0011428466532379389,
-0.013760843314230442,
0.032348036766052246,
0.016317659988999367,
-0.02829734981060028,
-0.029302839189767838,
0.03338225185871124,
0.01289181225001812,
-0.01219515223056078,
0.006320219952613115,
-0.02181912399828434,
-0.014529324136674404,
-0.024749407544732094,
-0.0019912284333258867,
-0.00963115319609642,
-0.007171295117586851,
-0.010241629555821419,
-0.016332022845745087,
-0.018687741830945015,
-0.002355718519538641,
-0.037490397691726685,
0.022149499505758286,
0.020310889929533005,
-0.012740989215672016,
-0.01033499650657177,
0.006108349189162254,
0.012231062166392803,

-0.004047095309942961,
0.014780696481466293,
-0.01832863874733448,
0.009616789408028126,
0.01946340501308441,
-0.02462013065814972,
-0.03114144876599312,
0.028598995879292488,
-0.01858719252049923,
0.019319763407111168,
0.004413381218910217,
0.002976967254653573,
-0.011268665082752705,
-0.010694099590182304,
0.03967374563217163,
-0.020942911505699158,
0.00510285934433341,
-0.016389479860663414,
-0.02181912399828434,
0.01927667111158371,
-0.005260865204036236,
0.0035192135255783796,
0.032606590539216995,
-0.01555635966360569,
0.0036700370255857706,
0.0012622484937310219,
-0.046740900725126266,
-0.02019597589969635,
-0.028728272765874863,
-0.013301190920174122,
-0.011541583575308323,
-0.030509427189826965,
0.010306267999112606,
-0.037002015858888626,
0.02295389026403427,
0.015024887397885323,
-0.005278820171952248,
-0.004908943548798561,
-0.011002928949892521,
0.009336688555777073,
0.009013495407998562,
0.017251327633857727,
0.02276715636253357,
-0.014809424988925457,
-0.008331199176609516,
0.005221363622695208,
-0.02249423786997795,
0.030480697751045227,

0.036772191524505615,
-0.0036951741203665733,
0.028024431318044662,
-0.003824451472610235,
-0.034244101494550705,
-0.021201465278863907,
-0.0026771160773932934,
0.0032193621154874563,
0.010694099590182304,
0.018299909308552742,
-0.018386095762252808,
-0.00801518838852644,
-0.035393234342336655,
0.004438518080860376,
0.004373879637569189,
-0.018615921959280968,
-0.003978865686804056,
-0.013394557870924473,
0.0005440416862256825,
-0.012259790673851967,
-0.026631109416484833,
0.010406817309558392,
-0.03651363402605057,
-0.018184997141361237,
0.0007451396086253226,
0.008525115437805653,
0.020770542323589325,
-0.016145288944244385,
-0.009121227078139782,
0.01066537108272314,
0.005835430696606636,
-0.00678705470636487,
-0.0009107760270126164,
0.009437237866222858,
0.006923513952642679,
0.035364504903554916,
0.020497623831033707,
0.03185965493321419,
0.0034545748494565487,
0.01946340501308441,
-0.010945471934974194,
0.019535226747393608,
0.0034348242916166782,
-0.024792499840259552,
-0.019492132589221,
-0.010945471934974194,
-0.011814502067863941,
0.018457915633916855,

0.0026124774012714624,
0.005900069139897823,
-0.0068696485832333565,
-0.004797621630132198,
0.010176991112530231,
-0.014795061200857162,
0.0038172693457454443,
-0.021503113210201263,
-0.04151235520839691,
-0.026975849643349648,
-0.0036448996979743242,
-0.027119489386677742,
0.021416928619146347,
-0.034014277160167694,
-0.01854410022497177,
0.05808856710791588,
-0.00856820773333311,
-0.011326122097671032,
0.001365490723401308,
0.024921776726841927,
-0.015470175072550774,
0.02206331305205822,
0.007426258642226458,
0.015628181397914886,
-0.026616744697093964,
0.015168528072535992,
-0.023284265771508217,
-0.02762223407626152,
-0.012489616870880127,
0.010507365688681602,
0.021890943869948387,
-0.005892887245863676,
0.010579186491668224,
-0.031428731977939606,
0.04668344557285309,
-0.007957731373608112,
-0.0007209001341834664,
-0.017955170944333076,
-0.025812353938817978,
-0.008503569290041924,
-0.01715077832341194,
0.014924338087439537,
-0.02858463115990162,
0.020353982225060463,
-0.001348433317616582,
0.014593962579965591,
0.008697484619915485,
-0.051222510635852814,

0.0016752174124121666,
0.007121020928025246,
-0.006611093878746033,
0.03763403743505478,
0.03800750523805618,
0.03016468696296215,
0.020368345081806183,
0.020986003801226616,
-0.01996614970266819,
0.005127996671944857,
-0.03295132890343666,
-0.0075196255929768085,
0.032807689160108566,
0.006952241994440556,
-0.012690714560449123,
-0.028656452894210815,
0.00713179400190711,
0.011060385033488274,
-0.020368345081806183,
-0.04036322236061096,
0.020152883604168892,
0.0020217522978782654,
0.007770997937768698,
-0.02828298509120941,
0.005874931812286377,
-0.027220038697123528,
0.01369620393961668,
-0.01834300346672535,
0.005411688704043627,
-0.014349772594869137,
-0.011498491279780865,
0.004219464957714081,
0.025927266106009483,
-0.0028836005367338657,
0.03708820044994354,
0.004553431179374456,
-0.016791675239801407,
-0.01090956199914217,
0.007598628289997578,
-0.017280057072639465,
0.00964551791548729,
0.017122050747275352,
0.048148587346076965,
-0.008252196945250034,
0.003479712177067995,
0.016332022845745087,
0.008561025373637676,
-0.017840256914496422,

-0.02693275548517704,
0.003558714874088764,
0.0375765822827816,
-0.033985547721385956,
0.011002928949892521,
0.009358234703540802,
-0.02576926164329052,
-0.0036125804763287306,
-0.024763772264122963,
-0.004621660802513361,
-0.014687329530715942,
-0.013847027905285358,
-0.0072897993959486485,
0.013746478594839573,
0.026975849643349648,
-0.012087421491742134,
-0.020612535998225212,
-0.013653111644089222,
-0.002630432602018118,
-0.027521686628460884,
-0.028426626697182655,
0.011771409772336483,
-0.041627269238233566,
-0.03559432923793793,
-0.014471868053078651,
-0.035192135721445084,
0.01603037677705288,
-0.008970403112471104,
0.01577182300388813,
-0.0009534195996820927,
0.045850325375795364,
-0.0057312906719744205,
0.029389023780822754,
-0.02880009450018406,
-0.026401283219456673,
-0.02365773357450962,
0.0010099783539772034,
-0.014550870284438133,
-0.004140462260693312,
0.0029213062953203917,
-0.012159241363406181,
-0.024433396756649017,
-6.665856926701963e-05,
0.0082450145855546,
0.01696404628455639,
-0.009128408506512642,
-0.03392809256911278,
0.001092572114430368,

0.023370450362563133,
0.012647622264921665,
-0.014543688856065273,
-0.01231006532907486,
0.05895041674375534,
-0.020957276225090027,
0.000309053371893242,
-0.006316629238426685,
-0.00661468505859375,
-0.02160366252064705,
0.003957319539040327,
0.022364960983395576,
0.0031026534270495176,
-0.007128202822059393,
-0.016174018383026123,
-0.002163598081097007,
-0.0034258465748280287,
-0.02278152108192444,
-0.03013595938682556,
-0.02648746781051159,
-0.039530105888843536,
-0.02737804502248764,
0.016533121466636658,
0.009516241028904915,
0.00504181208088994,
0.0011850412702187896,
0.0027615053113549948,
0.026042180135846138,
0.02878572978079319,
-0.023111896589398384,
-0.02041143737733364,
-0.035852886736392975,
-0.006180169992148876,
-0.002975171897560358,
-0.026990212500095367,
-0.015901099890470505,
0.018845748156309128,
0.039846114814281464,
0.00632740231230855,
-0.010069259442389011,
-0.019377220422029495,
-0.018299909308552742,
-0.021689847111701965,
-0.014909974299371243,
-0.007893092930316925,
0.013064182363450527,
-0.0011284825159236789,
-0.017294419929385185,

0.020152883604168892,
-0.003350434824824333,
0.020310889929533005,
-0.006126304157078266,
-0.02691839262843132,
0.006532091181725264,
0.02087108977138996,
0.0058857048861682415,
-0.02973376214504242,
-0.0006401018472388387,
-0.008769305422902107,
0.03200329840183258,
0.017955170944333076,
0.005666651763021946,
0.011340485885739326,
-0.005178271327167749,
-0.009710156358778477,
-0.037691496312618256,
-0.010442727245390415,
0.009078134782612324,
0.0007586059509776533,
0.013308372348546982,
-0.017524246126413345,
-0.032405491918325424,
0.005727699492126703,
0.006320219952613115,
0.020483259111642838,
-0.03838097304105759,
0.014313862659037113,
0.018730834126472473,
0.03461756929755211,
-0.03550814464688301,
-0.003233726369217038,
-0.006783463526517153,
-0.03295132890343666,
0.01578618586063385,
-0.006632640026509762,
8.281822374556214e-05,
0.0334971658885479,
-0.01217360608279705,
0.002307239454239607,
-0.012130513787269592,
0.0281249787658453,
-0.02482122741639614,
-0.0025622029788792133,
0.010837741196155548,
0.010018985718488693,
8.887809235602617e-05,

-0.009437237866222858,
-0.00014902792463544756,
-0.016576213762164116,
-0.011053203605115414,
-0.008467658422887325,
0.007634538691490889,
-0.010988564230501652,
-0.0021797576919198036,
-0.005239319056272507,
-0.043982986360788345,
-0.02993486076593399,
0.015010522678494453,
0.02206331305205822,
0.02762223407626152,
-0.008812397718429565,
-0.021876581013202667,
-0.005038220901042223,
0.007627356797456741,
0.012726625427603722,
-0.014227677136659622,
-0.027966974303126335,
-0.0046647535637021065,
0.003910636063665152,
-0.028670817613601685,
0.021732939407229424,
0.17558720707893372,
0.008079826831817627,
0.026746023446321487,
0.035393234342336655,
-0.011699588969349861,
0.03257786110043526,
0.006255581509321928,
-0.023557184264063835,
0.01043554488569498,
0.0352783203125,
0.030566882342100143,
0.0019517270848155022,
0.005440416745841503,
-0.01254707295447588,
0.0021653936710208654,
0.015613816678524017,
-0.028455354273319244,
-0.03266404569149017,
-0.02950393594801426,
-0.013509470969438553,
-0.012187969870865345,
-0.018170632421970367,
-0.017481153830885887,

-0.01463705487549305,
0.008762123063206673,
-0.0027489366475492716,
-0.015484539791941643,
0.005472735967487097,
-0.006090393755584955,
0.014112764038145542,
-0.008166011422872543,
-0.0034653479233384132,
0.01717950776219368,
-0.019879965111613274,
-0.019089937210083008,
-0.010406817309558392,
0.01888840451836586,
0.004316423088312149,
0.030337056145071983,
-0.00918586552143097,
0.02039707452058792,
-0.026559289544820786,
0.03088289313018322,
-0.009035041555762291,
0.013746478594839573,
0.02668856643140316,
-0.010751556605100632,
-0.017940806224942207,
-0.0024742225650697947,
-0.008970403112471104,
-0.02115837298333645,
0.0029159197583794594,
0.05737036094069481,
0.01880265586078167,
0.024893049150705338,
-0.01858719252049923,
0.010184172540903091,
0.009537787176668644,
0.011283029802143574,
0.003960910718888044,
-0.023528456687927246,
-0.0008798033813945949,
0.0008842921815812588,
0.02740677259862423,
-0.038151148706674576,
-0.010967018082737923,
-0.02625764161348343,
-0.012159241363406181,
-0.006399223115295172,
-0.0058282483369112015,
-0.0025604073889553547,

-0.03243422135710716,
0.012812810018658638,
-0.0029787628445774317,
-0.0015665886458009481,
-0.03272150456905365,
0.028211165219545364,
0.013301190920174122,
0.023097531870007515,
0.01791207864880562,
-0.008280924521386623,
-0.02041143737733364,
-0.016648033633828163,
-0.013997850939631462,
-0.014127128757536411,
-0.031227633357048035,
-0.0023305811919271946,
0.015944192185997963,
-0.0105432765558362,
-0.013911666348576546,
-0.030222143977880478,
-0.010622278787195683,
0.008051098324358463,
0.002608886221423745,
0.018860111013054848,
0.02295389026403427,
-0.002364696003496647,
0.024778135120868683,
0.0031870428938418627,
-0.01486688107252121,
-0.03904172405600548,
0.055129557847976685,
-0.0009426464675925672,
-0.008058280684053898,
-0.012446524575352669,
2.2934926164452918e-05,
0.010478638112545013,
-0.00869030226022005,
-0.013387375511229038,
-0.01811317726969719,
0.012934905476868153,
-0.02370082587003708,
-0.0004030935815535486,
0.0057348813861608505,
3.1337385735241696e-05,
0.013358647003769875,
0.014651419594883919,
-0.03125636279582977,
0.021014731377363205,

0.0077063594944775105,
0.022652244195342064,
-0.006862466223537922,
-0.004592932760715485,
0.008237832225859165,
-0.01336582936346531,
-0.003348639467731118,
-0.01713641546666622,
-0.0024526764173060656,
0.020483259111642838,
-0.01252552680671215,
0.008316835388541222,
-0.015197256579995155,
-0.0021941219456493855,
-0.027564778923988342,
-0.0011904278071597219,
0.014471868053078651,
-0.012618893757462502,
-0.004032731521874666,
0.0187882911413908,
-0.01906120963394642,
0.01440722867846489,
0.0027704827953130007,
-0.01694968156516552,
-0.014809424988925457,
-0.019908692687749863,
-0.014134310185909271,
0.01254707295447588,
-5.6025743106147274e-05,
-0.005472735967487097,
-0.019161758944392204,
-0.018529735505580902,
-0.023327358067035675,
0.0006369596812874079,
0.003925000317394733,
0.03717438504099846,
-0.026631109416484833,
-0.021732939407229424,
-0.03318115696310997,
-0.016202745959162712,
0.01577182300388813,
-0.037748951464891434,
0.011225572787225246,
0.028843186795711517,
-0.006539273075759411,
-0.049096617847681046,
-0.0034132779110223055,
-0.18546973168849945,

0.010859287343919277,
0.03651363402605057,
-0.02788078971207142,
0.02881445735692978,
0.005806702189147472,
0.016992773860692978,
-0.017782801762223244,
-0.011527219787240028,
-0.01883138343691826,
0.030078502371907234,
-0.023528456687927246,
-0.014593962579965591,
-0.011017292737960815,
-0.026272006332874298,
0.016662398353219032,
-0.011936597526073456,
0.012719443067908287,
0.03430155664682388,
0.009932800196111202,
0.033525895327329636,
-0.019089937210083008,
0.01947776973247528,
-0.006427951157093048,
0.0033989138901233673,
-0.02046889439225197,
-0.008367110043764114,
0.019176121801137924,
0.008122919127345085,
-0.013265280053019524,
0.0031241998076438904,
0.014098400250077248,
0.0019786597695201635,
-0.014256405644118786,
0.034473929554224014,
0.009149954654276371,
0.00032207087497226894,
-0.014780696481466293,
0.0039824568666517735,
0.023384815081954002,
-0.029173562303185463,
0.03206075355410576,
0.011311757378280163,
-0.02133074216544628,
-0.015283441171050072,
0.013861391693353653,
0.015455811284482479,
-0.012015600688755512,
0.008000824600458145,

-0.03622635453939438,
0.004298468120396137,
-0.029647577553987503,
-0.016432572156190872,
0.0017658910946920514,
0.013430467806756496,
0.01809881255030632,
-0.006047301460057497,
0.014026579447090626,
0.013408921658992767,
-0.021675482392311096,
0.0047258008271455765,
-0.03065306693315506,
0.0010791057720780373,
0.01101011037826538,
0.004704254679381847,
0.004614478908479214,
-0.03059561178088188,
0.02296825498342514,
0.003249885980039835,
0.013473560102283955,
-0.016116561368107796,
-0.01857282966375351,
0.03019341453909874,
0.011936597526073456,
0.016849132254719734,
0.009911254048347473,
0.002161802491173148,
0.02413174882531166,
0.010234447196125984,
-0.0009938187431544065,
-0.020296525210142136,
0.02898682840168476,
-0.01427076943218708,
0.007806908339262009,
0.023600276559591293,
-0.007153340149670839,
-0.027291860431432724,
-0.015369626693427563,
-0.007770997937768698,
0.016403844580054283,
0.03602525591850281,
-0.0030757207423448563,
-0.005953934509307146,
-0.014428775757551193,
0.01811317726969719,
0.005167498253285885,
0.0018233476439490914,

0.003616171423345804,
0.015915462747216225,
0.0017066389555111527,
-0.0029643988236784935,
-0.0016940704081207514,
-0.02667420171201229,
0.01600164920091629,
0.03685837611556053,
0.002723799319937825,
-0.01903248205780983,
0.030538154765963554,
0.024189205840229988,
-0.026056544855237007,
-0.02066999301314354,
0.019161758944392204,
0.003057765541598201,
0.014967430382966995,
0.000889678718522191,
0.03108399175107479,
0.016202745959162712,
-0.019592681899666786,
0.0027633009012788534,
-0.0003986047813668847,
0.04148362576961517,
-0.01577182300388813,
-0.014321044087409973,
-0.003763403743505478,
-0.0017829485004767776,
-0.008855490013957024,
-0.07365929335355759,
-0.04441390931606293,
0.006370494607836008,
0.007318527437746525,
-0.011814502067863941,
-0.01229570060968399,
0.0019660911057144403,
0.04720055311918259,
0.005472735967487097,
-0.02368646115064621,
0.0038675437681376934,
0.015599452890455723,
-0.0004987048450857401,
0.00353896408341825,
0.016102196648716927,
-0.029317203909158707,
-0.01692095212638378,
0.005961116869002581,
-0.02131637930870056,

0.0250510536134243,
0.009523422457277775,
-0.02364336885511875,
0.005914433393627405,
-0.0007823965861462057,
-0.001996614970266819,
0.024088656529784203,
-0.03384190425276756,
0.02016724832355976,
0.018156269565224648,
-0.002242600778117776,
0.012417796067893505,
-0.016389479860663414,
-0.0029446480330079794,
-0.014479049481451511,
0.006582365836948156,
0.012231062166392803,
0.006830147001892328,
-0.015872370451688766,
0.03315242752432823,
-0.01429231557995081,
-0.0028800093568861485,
0.03881189599633217,
0.011541583575308323,
-0.05863440781831741,
-0.01786898635327816,
0.004337969236075878,
-0.007207205519080162,
0.023772645741701126,
-0.01644693687558174,
-0.010227265767753124,
-0.04300622642040253,
0.020497623831033707,
-0.028871914371848106,
0.0033252977300435305,
0.006654186174273491,
-0.00941569171845913,
0.0031672921031713486,
0.016791675239801407,
0.002328785602003336,
-0.04056432098150253,
-0.004021958447992802,
-0.024217935279011726,
-0.0030398103408515453,
0.046108879148960114,
0.01598728448152542,
0.016073469072580338,
0.0029877403285354376,

-0.020813634619116783,
0.006553637329488993,
-0.021287649869918823,
0.007656084839254618,
0.009257686324417591,
-0.012187969870865345,
0.02572616934776306,
-0.029302839189767838,
0.004287694580852985,
-0.003792132018133998,
-0.013179095461964607,
0.0005862363032065332,
-0.008295289240777493,
-0.013703386299312115,
-0.00787872914224863,
-0.003082902869209647,
-0.0018978615989908576,
0.014852517284452915,
0.02970503456890583,
0.016877859830856323,
0.03266404569149017,
0.033583350479602814,
-0.03780640661716461,
0.01693531684577465,
-0.007483715191483498,
0.02671729400753975,
-0.033296067267656326,
-0.017538610845804214,
0.007056382019072771,
0.0276078712195158,
-0.017064593732357025,
-0.001650080201216042,
0.003989638760685921,
-0.06119122356176376,
-0.021029096096754074,
-0.08153083920478821,
0.02463449351489544,
0.0032732277177274227,
-0.04596523568034172,
-0.004323604982346296,
-0.030078502371907234,
0.007828454487025738,
-0.02410302124917507,
-0.005167498253285885,
0.017021501436829567,
-0.011792955920100212,
0.01736624166369438,
0.01417740248143673,

-0.007314936723560095,
-0.00999025721102953,
-0.02572616934776306,
0.028254257515072823,
-0.029762491583824158,
0.021258922293782234,
0.00615503266453743,
-0.0067762816324830055,
-0.01809881255030632,
0.012331611476838589,
0.010485819540917873,
-0.006751144304871559,
0.02598472312092781,
-0.002154620597139001,
0.023772645741701126,
0.00021478872804436833,
-0.021934036165475845,
0.006596729625016451,
-0.02368646115064621,
-0.010356542654335499,
0.016619306057691574,
0.01429231557995081,
0.0032319307792931795,
0.008661574684083462,
0.029992317780852318,
0.02713385410606861,
0.021215829998254776,
-0.01650439389050007,
-0.018874475732445717,
0.02388755977153778,
-0.029647577553987503,
-0.01969323121011257,
0.02230750396847725,
-0.011046021245419979,
0.01195096131414175,
0.020842362195253372,
-0.004546249285340309,
0.025625620037317276,
0.017797164618968964,
-0.03088289313018322,
-0.019865600392222404,
0.009085316210985184,
-0.029073012992739677,
0.014277951791882515,
0.014076854102313519,
-0.003285796381533146,
0.006639822386205196,
0.02533833682537079,

-0.00048434073687531054,
0.007735087536275387,
5.150889774085954e-05,
0.013279644772410393,
-0.012726625427603722,
0.0017102300189435482,
0.010758738033473492,
0.0008420975063927472,
-0.03151491656899452,
-0.00481557659804821,
-0.02576926164329052,
-0.003553328337147832,
0.019592681899666786,
0.028886279091238976,
0.0005440416862256825,
-0.011929415166378021,
-0.015168528072535992,
0.006769099272787571,
0.047028183937072754,
0.010241629555821419,
-0.0028638497460633516,
-0.013373011723160744,
0.01449341420084238,
0.0334971658885479,
0.012001235969364643,
-0.006172987632453442,
-0.0039034541696310043,
0.006101167295128107,
-0.02436157502233982,
-0.027306223288178444,
0.021244557574391365,
7.529949652962387e-05,
0.011132205836474895,
-0.0026771160773932934,
0.02065562829375267,
-0.01831427402794361,
0.012137695215642452,
0.014823788776993752,
-0.007235934026539326,
0.005013083573430777,
0.018903203308582306,
0.02528087981045246,
-0.035881612449884415,
-0.02668856643140316,
-0.0009152648271992803,
-0.03019341453909874,
0.0032983648125082254,
-0.010270358063280582,

0.018012627959251404,
0.009616789408028126,
-0.013193459250032902,
0.007052791304886341,
-0.0014319248730316758,
-0.007095883600413799,
0.021991493180394173,
-0.010399634949862957,
-0.012051510624587536,
-0.030825437977910042,
0.034014277160167694,
0.0038172693457454443,
0.007483715191483498,
0.04984355345368385,
-0.0032534769270569086,
0.0040973699651658535,
0.016533121466636658,
0.0003483303007669747,
-0.006348948460072279,
0.024792499840259552,
-0.000569178897421807,
0.008862672373652458,
0.004083005711436272,
-0.019578319042921066,
-0.02510851062834263,
-0.004086596891283989,
-0.019449040293693542,
-0.045160844922065735,
-0.014134310185909271,
0.004862260073423386,
0.09221775829792023,
0.0001850504777394235,
-0.0067798723466694355,
0.01521162036806345,
-0.02393065206706524,
0.022163862362504005,
0.015111071988940239,
0.011979689821600914,
-0.0050489939749240875,
-0.012683533132076263,
-0.009918436408042908,
-0.01623147539794445,
-0.018041355535387993,
-0.04803367331624031,
-0.02556816302239895,
0.010076441802084446,
-0.010011803358793259,
0.021632390096783638,

```

        0.0035407596733421087,
        -0.004973582457751036,
        0.030940350145101547,
        0.020842362195253372,
        0.0033540260046720505,
        0.005250092130154371,
        -0.026372555643320084,
        0.005189044401049614,
        0.02069872058928013,
        -0.007641720585525036,
        -0.004226647317409515,
        -0.005189044401049614,
        0.011642132885754108,
        -0.0027704827953130007,
        -0.01952086202800274,
        -0.011792955920100212,
        0.007699177134782076,
        0.00823064986616373,
        -0.0017416515620425344,
        -0.0022013038396835327,
        0.029647577553987503,
        -0.011096295900642872,
        0.0046288431622087955,
        0.021517476066946983,
        -0.039357732981443405,
        -0.017725344747304916,
        -0.0022318277042359114,
        -0.027478594332933426,
        0.0018287341808900237,
        -0.01102447509765625,
        -0.03682964667677879
    ],
}
},
{
    "_index": "github_issues",
    "_id": "C2LQUZMBKaR7I-arfsTE",
    "_score": 1.0,
    "_source": {
        "_type": "issue",
        "_repo": "langchain",
        "_issueNumber": "28267",
        "_title": "core[patch]: support final AIMessage responses in `tool_example_to_messages`",
        "_createdAt": "2024-11-21T20:47:32Z",
        "_closedAt": "2024-12-31T00:36:30Z",
        "_state": "open",
        "_body": "We have a test"
    }
}

```

[test_structured_few_shot_examples] (https://github.com/langchain-ai/langchain/blob/ad4333ca032033097c663dfe818c5c892c368bd6/libs/standard-tests/langchain_tests/integration_tests/chat_models.py#L546) in standard integration tests that implements a version of tool-calling few shot examples that works with ~all tested providers. The formulation supported by ~all providers is: `human message, tool call, tool message, AI response`.\r\n\r\nHere we update `langchain_core.utils.function_calling.tool_example_to_messages` to support this formulation.\r\n\r\nThe `tool_example_to_messages` util is undocumented outside of our API reference. IMO, if we are testing that this function works across all providers, it can be helpful to feature it in our guides. The structured few-shot examples we document at the moment require users to implement this function and can be simplified.",

```
"GitHub_Issue_vector": [  
    -0.03979385271668434,  
    0.005043520592153072,  
    0.01196878869086504,  
    -0.009051350876688957,  
    -0.01035690400749445,  
    0.009598370641469955,  
    0.0009235513862222433,  
    -0.012916956096887589,  
    -0.03947293385863304,  
    -0.020465826615691185,  
    0.0261402428150177,  
    -0.0029302015900611877,  
    0.004613198339939117,  
    -0.0009116993169300258,  
    -0.005616067908704281,  
    0.0010092511074617505,  
    0.010240206494927406,  
    0.0003307189326733351,  
    0.005561365745961666,  
    -0.01487163919955492,  
    0.011735393665730953,  
    0.03492173179984093,  
    -0.00236130109988153,  
    -0.0005260505131445825,  
    -0.008059421554207802,  
    0.007993779145181179,  
    0.024127209559082985,  
    -0.03856852650642395,  
    -0.015739576891064644,  
    0.003318585455417633,  
    0.018263161182403564,  
    0.005112809594720602,  
    -0.02285812422633171,  
    -0.008949240669608116,  
    -0.016541872173547745,
```

-0.030895665287971497,
0.023208217695355415,
0.01607508212327957,
0.040639907121658325,
-0.004536615684628487,
0.0269425380975008,
0.012829432263970375,
0.006535060703754425,
-0.017008662223815918,
-0.030516399070620537,
0.014040169306099415,
0.021559864282608032,
-0.026315288618206978,
-0.037518247961997986,
0.01715453341603279,
0.03200429305434227,
0.0002509452460799366,
-0.026913363486528397,
0.005831228569149971,
0.011538466438651085,
0.030458049848675728,
-0.003395168110728264,
0.0195905938744545,
0.0008579090354032815,
-0.00034690159372985363,
-0.02253720723092556,
0.01272002886980772,
-0.011210254393517971,
0.024273082613945007,
-0.012530394829809666,
-0.01576875150203705,
-0.017781782895326614,
0.014390261843800545,
0.012085486203432083,
0.003517335979267955,
0.0245356522500515,
0.030195480212569237,
0.020567936822772026,
-0.02382087893784046,
0.0293202493339777,
0.005809348076581955,
0.01484975777566433,
0.0010356904240325093,
-0.003931247163563967,
0.0013292576186358929,
0.03710980713367462,
-0.010466308332979679,
-0.005761939566582441,

0.029816213995218277,
0.015797926113009453,
0.008518918417394161,
-0.0011450942838564515,
0.018729951232671738,
0.0015945620834827423,
-0.012989891692996025,
0.02011573314666748,
-0.0038874857127666473,
0.0073993513360619545,
0.031595852226018906,
-0.005648888647556305,
0.010779932141304016,
-0.004168289247900248,
0.04723331704735756,
-0.0003143083304166794,
-0.009788003750145435,
-0.014725767076015472,
-0.007464993745088577,
-0.018058938905596733,
-0.008847130462527275,
-0.009824471548199654,
0.005750999320298433,
-0.00839492678642273,
-0.006768455728888512,
0.0019145684782415628,
0.016804441809654236,
-0.0054300809279084206,
0.038976967334747314,
-0.009824471548199654,
-0.027511438354849815,
-0.0012070898665115237,
0.0029520823154598475,
-0.00530973682180047,
-0.03705145791172981,
-0.009591076523065567,
-0.015126914717257023,
0.01747545227408409,
0.012712734751403332,
0.0023357735481113195,
-0.013325396925210953,
0.02936401031911373,
0.011793741956353188,
-0.01603132113814354,
-0.019838577136397362,
-0.04148596525192261,
-0.03865605220198631,
0.012464752420783043,

-0.0058494629338383675,
0.01969270408153534,
-0.006549648009240627,
-0.0015571824042126536,
0.024652348831295967,
-0.005134690552949905,
-0.018481967970728874,
-0.012027136981487274,
0.0014660124434158206,
-0.019721878692507744,
0.025790149345993996,
-0.01296071708202362,
0.014842464588582516,
-0.04603716731071472,
0.02065545879304409,
0.023645833134651184,
0.0017139946576207876,
0.02544005773961544,
-0.021807847544550896,
0.011545759625732899,
-0.0309248398989439,
-0.004649666603654623,
-0.00677939597517252,
0.013040946796536446,
0.0238938145339489,
0.010604886338114738,
0.026184003800153732,
-0.01711077243089676,
-0.0045183817856013775,
0.007140428759157658,
0.025527579709887505,
0.01457260176539421,
-0.004824712872505188,
0.013930764980614185,
0.030633095651865005,
0.030633095651865005,
-0.02490033209323883,
-0.011290484108030796,
-0.014390261843800545,
-0.022347573190927505,
0.00011117033136542886,
-0.01797141693532467,
0.0142589770257473,
0.0015097740106284618,
0.004412624519318342,
0.03976467624306679,
-0.0007963693351484835,
-0.0028080339543521404,

-0.010925804264843464,
0.02679666504263878,
0.017986003309488297,
0.025425469502806664,
0.03390062600374222,
-0.012362642213702202,
0.010954978875815868,
0.010349610820412636,
-0.03576778620481491,
0.012654386460781097,
0.00677939597517252,
0.04084412753582001,
0.026650793850421906,
0.0018115465063601732,
0.004711661953479052,
-0.5755521059036255,
-0.020101146772503853,
-0.026213178411126137,
0.005787467118352652,
0.028970157727599144,
0.027905292809009552,
-0.013376452028751373,
-0.00671010697260499,
-0.028007403016090393,
0.04119422286748886,
-0.03066227026283741,
0.016585633158683777,
0.0045074415393173695,
-0.01849655620753765,
-0.012515808455646038,
-0.003796315984800458,
0.0006295283674262464,
-0.027088409289717674,
0.0031216584611684084,
0.01737334206700325,
-0.04921717569231987,
0.024944093078374863,
0.022580968216061592,
-0.020188668742775917,
-0.009058644063770771,
-0.023251978680491447,
0.010181858204305172,
-0.02345619909465313,
0.014434023760259151,
-0.0067502218298614025,
-0.019605182111263275,
0.001269085449166596,
0.014630950056016445,

-0.003052369225770235,
0.04904212802648544,
0.012049018405377865,
-0.04002724587917328,
-0.015579117462038994,
0.0129096619784832,
6.5065414673881605e-06,
-0.02571721374988556,
0.005145630799233913,
0.006407422479242086,
0.0030231948476284742,
0.01893417164683342,
-0.0032018879428505898,
0.01651269756257534,
0.009620251134037971,
0.01010892167687416,
-0.02053876221179962,
0.008234468288719654,
-0.02355830930173397,
-0.004106293432414532,
0.0022537207696586847,
0.0035574506036937237,
0.005426434334367514,
0.03722650557756424,
-0.036555495113134384,
0.01492269430309534,
-0.006505886092782021,
-0.034454941749572754,
-0.010393371805548668,
-0.01466012466698885,
-0.008723138831555843,
-0.01236993633210659,
0.009000295773148537,
0.02560051716864109,
0.0017395222093909979,
-0.0027296277694404125,
-0.04764175787568092,
0.01331080961972475,
0.023470787331461906,
-0.001368460594676435,
-0.02484198287129402,
-0.010860161855816841,
0.019328026100993156,
-0.004985171835869551,
0.004321454558521509,
-0.025936022400856018,
0.018306922167539597,
0.011560346931219101,

0.0010201914701610804,
-0.01791306771337986,
-0.01591462269425392,
0.017927655950188637,
0.0027496851980686188,
0.006859625689685345,
-0.005874990485608578,
-0.0032675303518772125,
-0.008825249038636684,
0.012311587110161781,
0.027205107733607292,
-0.019867751747369766,
-0.04075660556554794,
-0.007687448523938656,
0.028445018455386162,
-0.00930662639439106,
-0.01296071708202362,
0.028167862445116043,
-0.03632210195064545,
-0.023747943341732025,
0.02457941323518753,
0.016643982380628586,
0.008008366450667381,
0.025615103542804718,
-0.008628321811556816,
-0.027205107733607292,
0.01124672219157219,
0.0024506477639079094,
-0.009153461083769798,
0.01339833252131939,
0.004974231589585543,
-0.005699944216758013,
-0.023106107488274574,
-0.01819022372364998,
-0.031158234924077988,
0.01623554155230522,
0.007950018160045147,
0.013339984230697155,
-0.003559274133294821,
-1.9302384316688403e-05,
-0.01715453341603279,
-0.013040946796536446,
0.0334630124270916,
0.035476043820381165,
0.014514252543449402,
0.0076436870731413364,
-0.03868522495031357,
0.006954442244023085,

0.025323359295725822,
-0.008022953756153584,
-0.021486928686499596,
0.021020138636231422,
-0.023310327902436256,
0.021311882883310318,
0.000345989887136966,
0.028795110061764717,
-0.009941169060766697,
0.019065456464886665,
-0.008927359245717525,
0.00012490279914345592,
0.01889040879905224,
0.015301961451768875,
-0.03807256370782852,
-0.013332691043615341,
-0.04061073437333107,
-0.0315375030040741,
0.0013037299504503608,
0.018802886828780174,
-0.009693186730146408,
0.003674148116260767,
-0.018248572945594788,
-0.0078114396892488,
0.0023540076799690723,
-0.00895653385668993,
-0.012100073508918285,
-0.00034393856185488403,
-0.029787039384245872,
-0.002671279013156891,
-0.01779637113213539,
0.00570359081029892,
0.029130615293979645,
0.001527096377685666,
0.03778081759810448,
-0.021107662469148636,
0.005743705667555332,
-0.02033454179763794,
0.03915201500058174,
-0.014995629899203777,
-0.03498007729649544,
-0.005732765421271324,
-0.030808143317699432,
-0.02609647996723652,
0.02436060458421707,
-0.0019528599223122,
0.006815863773226738,
-0.03130410611629486,

0.014995629899203777,
-0.015797926113009453,
-0.02017408236861229,
-0.0009290215675719082,
0.012078192085027695,
-0.02592143416404724,
0.014397555030882359,
0.004408977925777435,
0.013463974930346012,
0.021166011691093445,
0.00671010697260499,
-0.014638244174420834,
0.012997184880077839,
-0.03807256370782852,
0.018044352531433105,
-0.033813104033470154,
0.02743850275874138,
-0.025585928931832314,
0.025425469502806664,
-0.004923176020383835,
-0.010962272062897682,
-0.004026064183562994,
0.03500925377011299,
0.016906552016735077,
0.027511438354849815,
0.014280857518315315,
-0.0038838391192257404,
-0.01517067663371563,
-0.032937873154878616,
-0.003378757508471608,
-0.026227764785289764,
0.005692650564014912,
-0.006502239499241114,
-0.003590271808207035,
-0.024273082613945007,
-0.007614512462168932,
0.003499101847410202,
0.0038838391192257404,
-0.014981042593717575,
0.011677044443786144,
-0.020640872418880463,
-0.011837503872811794,
-0.0014705709181725979,
-0.011523879133164883,
-0.013784893788397312,
0.005014345981180668,
-0.014295444823801517,
0.013252461329102516,

-0.00986823346465826,
0.005550425499677658,
0.01715453341603279,
0.021384818479418755,
0.00874501932412386,
0.0018817473901435733,
0.0036285631358623505,
0.005758292973041534,
0.008548093028366566,
0.02549840696156025,
0.021822433918714523,
0.028445018455386162,
-0.012566863559186459,
0.06068270653486252,
-0.026592444628477097,
-0.021341057494282722,
-0.02403968758881092,
0.004339688457548618,
-0.010882043279707432,
0.053972598165273666,
0.00020706970826722682,
0.03258778154850006,
0.019605182111263275,
0.00017345079686492682,
-0.006892446894198656,
-0.003088837256655097,
-0.027248868718743324,
-0.002474352018907666,
-0.003076073480769992,
-0.0009153460850939155,
0.015301961451768875,
-0.008176119066774845,
0.0037124394439160824,
0.024010512977838516,
0.02393757738173008,
0.0024014159571379423,
0.008876304142177105,
0.022172527387738228,
-0.010852868668735027,
0.0226393174380064,
-0.011465530842542648,
0.017927655950188637,
-0.021909957751631737,
-0.03926871344447136,
0.0059515731409192085,
-0.003528276225551963,
-0.016162605956196785,
-0.006877859588712454,

-0.01807352714240551,
0.049946535378694534,
0.013194112107157707,
0.012406404130160809,
0.010400665923953056,
0.020772157236933708,
-0.015039391815662384,
-0.041865233331918716,
-0.03544686734676361,
0.020567936822772026,
-0.012873194180428982,
0.0007421232294291258,
-0.021209772676229477,
-0.02764272317290306,
0.025936022400856018,
-0.036234576255083084,
-0.01255956944078207,
-0.0026621620636433363,
-0.006597056053578854,
-0.026986299082636833,
0.008030247874557972,
0.0036103292368352413,
0.0211951844394207,
-0.0052331541664898396,
-0.009664013050496578,
0.00846786331385374,
-0.0012836726382374763,
-0.014784116297960281,
0.015374897047877312,
-0.009569196030497551,
0.014798702672123909,
0.021341057494282722,
-0.006225083023309708,
0.002235486637800932,
-0.03247108310461044,
-0.015389484353363514,
-0.0032055347692221403,
-0.01226782612502575,
-0.0015599174657836556,
0.0056598293595016,
-0.01619177870452404,
0.017723433673381805,
0.015214437618851662,
-0.029232725501060486,
0.010765345767140388,
0.04244871810078621,
-0.00890547875314951,
-0.009780710563063622,

-0.03054557368159294,
-0.010531950742006302,
0.014193334616720676,
0.01126860361546278,
0.0439949631690979,
0.014703886583447456,
-0.011480117216706276,
-0.035738613456487656,
-0.011429062113165855,
-0.05400177463889122,
-0.03588448464870453,
0.016439761966466904,
-0.009423323906958103,
-0.01554994285106659,
-0.003993242979049683,
0.005276915617287159,
-0.01086745597422123,
0.03629292547702789,
0.020961789414286613,
-0.0024287670385092497,
-7.322085002670065e-05,
-0.0011934144422411919,
-0.005488429684191942,
0.0012909661745652556,
0.009919288568198681,
0.006801276933401823,
0.01705242320895195,
0.014915400184690952,
0.01759214885532856,
0.011655163951218128,
0.029670342803001404,
-0.007300888188183308,
-0.04819607362151146,
0.001574504654854536,
0.00633448688313365,
0.011421768926084042,
0.03156667575240135,
-0.04151513800024986,
0.030458049848675728,
0.004697074647992849,
0.012070898897945881,
0.005550425499677658,
0.007326415739953518,
-0.005178452003747225,
0.001592738670296967,
-0.005889577325433493,
0.006068270653486252,
-0.007731209974735975,

-0.014696592465043068,
-0.0008957445388659835,
-0.003650444094091654,
0.006939854938536882,
-0.0281970351934433,
0.00047818629536777735,
-0.0066699921153485775,
-0.04139844328165054,
-0.024302257224917412,
0.032616954296827316,
0.031975116580724716,
-0.019269676879048347,
0.009999518282711506,
-0.02399592474102974,
-0.01517067663371563,
-0.0033149386290460825,
-0.016600221395492554,
0.004201110452413559,
-0.012588744051754475,
-0.016337651759386063,
-0.015885448083281517,
-0.039677154272794724,
0.0010402488987892866,
-0.02560051716864109,
0.010721583850681782,
-0.0012544982600957155,
-0.015156089328229427,
-0.028415843844413757,
0.002793446648865938,
0.013974526897072792,
0.020524173974990845,
0.024389779195189476,
-0.0019291556673124433,
0.013566086068749428,
0.008110476657748222,
-0.023208217695355415,
-0.01061947364360094,
0.018263161182403564,
-0.017067011445760727,
-0.012165715917944908,
-0.0019236855441704392,
-0.008263642899692059,
-0.011188373900949955,
-0.020144907757639885,
0.026315288618206978,
-0.02221628837287426,
0.0015234495513141155,
-0.03238355740904808,

0.0011159199057146907,
0.00951814092695713,
0.014434023760259151,
0.022785188630223274,
0.02420014701783657,
-0.004646019544452429,
-0.00720607116818428,
0.011429062113165855,
-0.02409803681075573,
-0.025629689916968346,
-0.03865605220198631,
-0.017387928441166878,
0.0036942055448889732,
0.008577266708016396,
0.03072061948478222,
-0.013164937496185303,
0.015637466683983803,
0.021866196766495705,
-0.020159494131803513,
0.01145823672413826,
-0.004587670788168907,
-0.0035683910828083754,
-0.005721824709326029,
0.0245356522500515,
0.011625989340245724,
0.002467058366164565,
-0.029816213995218277,
-0.00039886843296699226,
-0.05837792903184891,
0.031712546944618225,
-0.006961735896766186,
-0.03673054277896881,
0.03498007729649544,
0.02017408236861229,
-0.025089964270591736,
-0.026767492294311523,
-0.01967811770737171,
-0.013427507132291794,
0.0035337465815246105,
0.02743850275874138,
-0.01591462269425392,
-0.01185938436537981,
-0.02055334858596325,
0.0005862226244062185,
0.000770385900977999,
0.0032474729232490063,
0.004999759141355753,
0.013515030033886433,

-0.005036226939409971,
-0.021880783140659332,
-0.028430430218577385,
-0.008358458988368511,
-0.03255860507488251,
-0.012348054908216,
0.03775164484977722,
0.013624434359371662,
0.0160896684974432,
0.0076655675657093525,
0.00796460546553135,
-0.00965671893209219,
-0.03810173645615578,
-0.02490033209323883,
-0.01061947364360094,
0.0054191406816244125,
0.00044764435733668506,
0.019867751747369766,
-0.0022117826156318188,
0.04857533797621727,
-0.03582613542675972,
-0.01021832600235939,
0.004569436889141798,
0.0026202236767858267,
0.009401443414390087,
0.018642427399754524,
0.003426166018471122,
0.000989193795248866,
-0.023281153291463852,
0.009503553621470928,
0.0017832837766036391,
0.0015699461800977588,
-0.010590299032628536,
0.0004339688748586923,
-0.0018498378340154886,
-0.0017513743368908763,
0.028970157727599144,
-0.011516585946083069,
-0.01765049807727337,
-0.024389779195189476,
-0.003417049068957567,
-0.02323739230632782,
-0.008198000490665436,
-0.04805019870400429,
-0.00012923337635584176,
0.04023146629333496,
0.0035246293991804123,
0.009693186730146408,

0.03180007264018059,
0.030487224459648132,
-0.008511624298989773,
-0.007341002579778433,
-0.012224064208567142,
0.005765586160123348,
-0.024127209559082985,
-0.00034576194593682885,
-0.03778081759810448,
-0.004011476878076792,
0.003185477340593934,
0.00825634878128767,
0.021180598065257072,
0.023003997281193733,
0.010495482012629509,
-0.014835171401500702,
0.01657104678452015,
-0.0497131384909153,
0.008176119066774845,
-0.01667315699160099,
-0.024812808260321617,
-0.031449977308511734,
-0.006298018619418144,
0.005007052794098854,
-0.03480503335595131,
0.012472046539187431,
-0.0022646610159426928,
0.002441530814394355,
0.011844797059893608,
-0.03136245533823967,
-0.012501221150159836,
0.003898426191881299,
0.012800258584320545,
0.02592143416404724,
0.01925508864223957,
0.003814549883827567,
0.021880783140659332,
0.0055941869504749775,
-0.0031052478589117527,
0.005842169281095266,
0.002472528489306569,
0.009430618025362492,
0.022508032619953156,
0.02297482267022133,
-0.0022537207696586847,
-0.01417145412415266,
0.007413938641548157,
0.017840132117271423,

-0.0484294667840004,
-0.03372558206319809,
0.02560051716864109,
0.016921138390898705,
0.015856273472309113,
-0.02952446974813938,
0.0026731023099273443,
-0.03419237211346626,
0.017519213259220123,
-0.007526989560574293,
8.222387987188995e-05,
0.02109307423233986,
-0.0008574532112106681,
-0.006363661028444767,
0.03034135326743126,
0.025936022400856018,
0.034513287246227264,
-0.0007991044549271464,
-0.004868474323302507,
0.014754941686987877,
-0.009532728232443333,
-0.004554849583655596,
0.014630950056016445,
-0.00941603071987629,
0.036555495113134384,
-0.009372268803417683,
-0.00935768149793148,
0.037080634385347366,
-0.006024509202688932,
-0.00572911836206913,
-0.009014883078634739,
-0.0010320435976609588,
0.04084412753582001,
-0.03340466320514679,
0.009904701262712479,
0.006837744731456041,
-0.02059711143374443,
0.021282708272337914,
-0.04250706732273102,
0.002328480128198862,
-0.021982893347740173,
-0.034017324447631836,
0.004274046514183283,
0.037839166820049286,
0.004496500827372074,
0.008737726137042046,
-0.00500340573489666,
0.005707237403839827,

0.0004663341969717294,
0.02291647344827652,
-0.011560346931219101,
0.020932616665959358,
-0.022872712463140488,
-0.020363716408610344,
-0.012515808455646038,
-0.013653608970344067,
0.01261062454432249,
-0.012413697317242622,
-0.021151423454284668,
0.009029469452798367,
0.05216378718614578,
-0.028868047520518303,
0.010087041184306145,
-0.0159292109310627,
-0.011305071413516998,
-0.035738613456487656,
-0.003916660323739052,
-0.008562679402530193,
-0.008577266708016396,
0.017898481339216232,
-0.029334837570786476,
-0.02898474410176277,
-0.017767196521162987,
0.03769329562783241,
0.017708847299218178,
-0.005145630799233913,
0.006338133476674557,
-0.005889577325433493,
-0.004580377135425806,
0.012953423894941807,
-0.006556941196322441,
0.015739576891064644,
0.015331135131418705,
-0.030253829434514046,
-0.005331617314368486,
0.031187409535050392,
-0.013850536197423935,
-0.014382967725396156,
-0.011100850999355316,
0.012348054908216,
-0.009729654528200626,
-0.03147915378212929,
0.011151906102895737,
0.01693572662770748,
-0.023091519251465797,
-0.030166305601596832,

-0.022332986816763878,
-0.01070699654519558,
-0.013405626639723778,
-0.018365271389484406,
-0.00941603071987629,
0.013099296018481255,
0.0031508328393101692,
-0.0008807015256024897,
0.04320725426077843,
0.020188668742775917,
-0.011589521542191505,
-0.01775260828435421,
-0.007326415739953518,
-0.020319953560829163,
0.01145823672413826,
0.0045183817856013775,
-0.012865900993347168,
-0.009408736601471901,
0.02485656924545765,
-0.0015599174657836556,
-0.004941410385072231,
-0.004729895852506161,
-0.014302738942205906,
-0.027453089132905006,
-0.025644278153777122,
-0.025410883128643036,
0.005831228569149971,
0.004821065813302994,
-0.002611106727272272,
-0.013325396925210953,
0.006626230664551258,
-0.010444426909089088,
0.012253238819539547,
-0.017781782895326614,
-0.006469418294727802,
-0.005543131846934557,
0.0024032392539083958,
0.022610142827033997,
-0.0019875045400112867,
-0.035417694598436356,
-0.005991687998175621,
0.008285523392260075,
-0.004259459208697081,
0.010772638954222202,
0.026927949860692024,
0.0129096619784832,
-0.0218370221555233,
-0.013945352286100388,

-0.009408736601471901,
-0.005568659398704767,
-0.00598439434543252,
-0.0026329876855015755,
0.001115008257329464,
-0.015331135131418705,
0.020611697807908058,
-0.01851114258170128,
-0.004295927006751299,
-0.011560346931219101,
0.005364438518881798,
0.016425173729658127,
0.009481673128902912,
-0.01011621579527855,
-0.0012462929589673877,
0.004201110452413559,
-0.030137132853269577,
0.022245462983846664,
0.012654386460781097,
0.0018689836142584682,
-0.015126914717257023,
0.002913790987804532,
0.023003997281193733,
-0.0004927734844386578,
0.03565108776092529,
-0.008489743806421757,
-0.00044399756006896496,
0.014609069563448429,
0.002882793080061674,
0.0038656049873679876,
-0.009343094192445278,
0.006731987465173006,
-0.024666937068104744,
0.006648111157119274,
0.006050036754459143,
-0.01414957270026207,
-0.007840613834559917,
0.0012389993062242866,
-0.00976612325757742,
-0.023602072149515152,
-0.0016583810793235898,
-0.00874501932412386,
-0.0030250183772295713,
-0.0012526748469099402,
-0.020684633404016495,
0.016658568754792213,
-0.00142589770257473,
-0.0031690667383372784,

0.010758051648736,
0.0007959135109558702,
-0.011356126517057419,
0.008453276008367538,
0.006370954681187868,
-0.026607032865285873,
-0.0016829968662932515,
0.18636591732501984,
-0.02554216794669628,
0.007446759846061468,
0.035913657397031784,
-0.03168337419629097,
0.016804441809654236,
0.0003115732397418469,
-0.0019729172345250845,
0.0017769018886610866,
0.025673452764749527,
-0.0002370418224018067,
0.011013327166438103,
-0.01683361642062664,
-0.011436356231570244,
-0.014696592465043068,
0.0014113105135038495,
-0.016556458547711372,
-0.010254793800413609,
-0.018817473202943802,
0.018204811960458755,
0.016804441809654236,
-0.008774193935096264,
0.0001449601841159165,
-0.02075756900012493,
0.022085003554821014,
-0.011793741956353188,
-0.012727322056889534,
0.0013958116760477424,
0.005083635449409485,
-0.005017993040382862,
-0.025483818724751472,
0.012690854258835316,
0.008103183470666409,
0.004872120916843414,
0.01277108397334814,
-0.004463679622858763,
0.021472342312335968,
-0.0004351084935478866,
0.006060977000743151,
-0.005306089762598276,
0.028226209804415703,

-0.041544314473867416,
0.006473064888268709,
-0.030312178656458855,
0.003909366670995951,
0.03136245533823967,
-0.015447832643985748,
-0.0242147333920002,
-0.009547315537929535,
0.016746092587709427,
-0.003913013264536858,
0.026534097269177437,
0.010043279267847538,
0.022026654332876205,
0.03054557368159294,
-0.01769426092505455,
-0.021122248843312263,
0.006093798205256462,
0.009671306237578392,
0.02651950903236866,
-0.0357094369828701,
0.01576875150203705,
0.0033149386290460825,
0.01709618605673313,
-0.02695712447166443,
0.022070417180657387,
-0.012063604779541492,
-0.02259555459022522,
-0.020451238378882408,
0.006418363191187382,
0.004926823079586029,
-0.02910144254565239,
0.0058822836726903915,
0.0024488242343068123,
-0.02775941975414753,
-0.05017992854118347,
0.018379857763648033,
0.006910680793225765,
0.01080910675227642,
0.023470787331461906,
0.006757515016943216,
-0.037080634385347366,
0.001423162641003728,
0.007169603370130062,
-0.02221628837287426,
-0.020961789414286613,
0.008869010955095291,
0.005134690552949905,
0.009547315537929535,

-0.006939854938536882,
-0.008569973520934582,
-0.007614512462168932,
-0.004408977925777435,
0.025162899866700172,
0.023062344640493393,
0.017679672688245773,
0.007359236944466829,
0.009189928881824017,
-0.00900758896023035,
-0.015039391815662384,
-0.06348344683647156,
0.038160085678100586,
-0.018525728955864906,
0.002000268315896392,
-0.00010672579810488969,
-0.00353009975515306,
-0.01861325278878212,
-0.007898963056504726,
-0.017169121652841568,
-0.01677526719868183,
0.0019656235817819834,
-0.035096775740385056,
-0.006374601740390062,
-0.005565012339502573,
0.0009285657433792949,
-0.01765049807727337,
0.025731801986694336,
-0.014842464588582516,
0.026052718982100487,
-0.005149277858436108,
0.002720510819926858,
0.007235245779156685,
0.006834098137915134,
0.04346982389688492,
-0.0017668732907623053,
-0.027774007990956306,
-0.05362250655889511,
0.005466549191623926,
-0.00021162819757591933,
-0.0030724266543984413,
0.02544005773961544,
0.012807551771402359,
0.013675489462912083,
-0.02065545879304409,
-0.01967811770737171,
0.005736412014812231,
-0.0036942055448889732,

-0.014375674538314342,
0.012092779390513897,
0.004817419219762087,
0.00037926691584289074,
0.00018530288070905954,
-0.007191483862698078,
0.0054082004353404045,
0.026811253279447556,
-0.010699703358113766,
-0.010079747065901756,
0.017256643623113632,
-0.020611697807908058,
-0.0010037809843197465,
-0.03734320402145386,
0.0013146704295650125,
0.004591317847371101,
-0.009080525487661362,
0.02097637765109539,
-0.021268121898174286,
-0.017402516677975655,
-0.028649238869547844,
0.007337355986237526,
-0.020947203040122986,
-0.04735001549124718,
-0.0010657764505594969,
0.01749003864824772,
0.011370713822543621,
-0.036117877811193466,
-0.006793983280658722,
-0.1868327111005783,
0.02996208518743515,
0.023485373705625534,
-0.04195275530219078,
0.024331431835889816,
-0.011538466438651085,
0.007227952126413584,
-0.020305367186665535,
-0.01769426092505455,
-0.01110814418643713,
0.03594283387064934,
-0.004131821449846029,
-0.02473987266421318,
0.009131580591201782,
-0.007723916321992874,
0.014025582000613213,
-0.02049500122666359,
0.011071676388382912,
0.021443167701363564,

0.03398815169930458,
0.04831276834011078,
-0.024929504841566086,
0.02215793915092945,
0.016643982380628586,
0.01667315699160099,
-0.01011621579527855,
0.0002823988616000861,
0.0134712690487504,
-0.0025837558787316084,
-0.01963435672223568,
-0.017242057248950005,
0.01452154666185379,
-0.0022865417413413525,
-0.034338243305683136,
0.03865605220198631,
0.0013985467376187444,
0.020042797550559044,
-0.03206264227628708,
-0.011305071413516998,
0.016045907512307167,
0.018117288127541542,
0.007322768680751324,
-0.02011573314666748,
-0.001725846785120666,
-0.022420508787035942,
0.006753868423402309,
0.02409803681075573,
-0.012413697317242622,
0.01737334206700325,
-0.010590299032628536,
0.010189151391386986,
-0.023645833134651184,
-0.0008583649178035557,
0.010641354136168957,
0.016366826370358467,
0.002417826559394598,
-0.011020621284842491,
0.0238938145339489,
0.014492372050881386,
-0.0052368007600307465,
0.010123508982360363,
-0.02333950251340866,
0.009299333207309246,
0.0028135040774941444,
0.013040946796536446,
-0.003438929794356227,
-0.03471750766038895,

0.03054557368159294,
0.010240206494927406,
0.013814067468047142,
-0.01845279335975647,
-0.010079747065901756,
0.01215842179954052,
-0.015535356476902962,
0.04481184482574463,
0.037897516041994095,
0.001454160432331264,
0.013347277417778969,
0.009189928881824017,
-0.004737189505249262,
-0.025688039138913155,
0.03818926215171814,
-0.014390261843800545,
0.03608870506286621,
-0.014747647568583488,
-0.007402998395264149,
-0.005543131846934557,
0.009248278103768826,
-0.04087330400943756,
0.0025254071224480867,
0.02657785825431347,
-0.008562679402530193,
-0.005138337146490812,
-0.009240983985364437,
0.01995527371764183,
0.010772638954222202,
0.005455608479678631,
-0.006793983280658722,
0.009576489217579365,
0.0056598293595016,
0.01417145412415266,
0.05012157931923866,
0.011006033979356289,
-0.0006938031874597073,
0.0010530126746743917,
-0.006370954681187868,
-0.03535934537649155,
0.017008662223815918,
0.04895460605621338,
-0.01645434834063053,
-0.017387928441166878,
0.013843242079019547,
-0.0020932615734636784,
0.016702331602573395,
-0.01527278684079647,

0.022449683398008347,
0.011582228355109692,
-0.01867160201072693,
0.011538466438651085,
-0.012756496667861938,
0.04805019870400429,
-0.02033454179763794,
-0.02339785173535347,
0.016921138390898705,
-0.0360303558409214,
0.002514466643333435,
-0.11062923818826675,
-0.008081302978098392,
-0.0022117826156318188,
0.0016337651759386063,
-0.015695815905928612,
-0.0024342371616512537,
-9.396200766786933e-05,
0.03500925377011299,
-0.036672193557024,
0.01379948016256094,
-0.013850536197423935,
-0.003112541511654854,
-0.03454246371984482,
0.003619446186348796,
0.01957600750029087,
-0.016104256734251976,
-0.029991259798407555,
-0.005313383415341377,
-0.020480412989854813,
0.04901295527815819,
-0.003362347139045596,
-0.019999036565423012,
0.02738015353679657,
-0.010123508982360363,
0.004624139051884413,
0.021880783140659332,
-0.024564826861023903,
0.016483522951602936,
0.02915978990495205,
0.014040169306099415,
0.027248868718743324,
-0.046241387724876404,
0.014463197439908981,
-0.008074008859694004,
0.0023722415789961815,
0.0055832467041909695,
-0.007421232294291258,

-0.0202324315905571,
0.03320044279098511,
-0.025571342557668686,
0.0034352829679846764,
0.012625211849808693,
-0.010283968411386013,
-0.025892259553074837,
-0.0020148553885519505,
-0.007169603370130062,
-0.022012067958712578,
0.018423618748784065,
0.008241761475801468,
-0.007723916321992874,
-0.04072743281722069,
0.016848202794790268,
-0.026855014264583588,
-0.013617141172289848,
0.03565108776092529,
0.007144075818359852,
-0.0014578071422874928,
0.0008063980494625866,
-0.008227174170315266,
-0.028488779440522194,
0.00427769310772419,
-0.018700776621699333,
0.000107694482721854,
0.01851114258170128,
0.021122248843312263,
-0.0032602366991341114,
-0.018029766157269478,
-0.0024925859179347754,
0.004339688457548618,
-0.015637466683983803,
0.0037324968725442886,
0.0369931124150753,
-0.013814067468047142,
0.0033459365367889404,
-0.01457260176539421,
-0.001506127300672233,
0.012165715917944908,
-0.01159681472927332,
-0.022887298837304115,
0.006593409460037947,
-1.1374872883607168e-05,
-0.007592631969600916,
-0.014689299277961254,
-0.01619177870452404,
0.00023248331854119897,

0.015579117462038994,
0.017125358805060387,
-0.01000681146979332,
0.012690854258835316,
-0.02920355275273323,
0.013938059099018574,
0.0038583113346248865,
0.007366530131548643,
0.006728340871632099,
0.019459309056401253,
0.029539057984948158,
0.0027588021475821733,
0.017183708027005196,
0.019065456464886665,
0.004923176020383835,
-0.07124383002519608,
-0.0006236023036763072,
-0.057794440537691116,
0.036117877811193466,
0.011523879133164883,
-0.01409851759672165,
-0.02021784335374832,
-0.02506078965961933,
0.012617918662726879,
-0.02850336767733097,
0.007173249963670969,
0.038218434900045395,
0.030166305601596832,
0.024725284427404404,
-0.011480117216706276,
-0.005090929102152586,
-0.015477007254958153,
-0.008074008859694004,
0.04247789457440376,
-0.025089964270591736,
0.027307217940688133,
0.00846786331385374,
0.013617141172289848,
0.0004166465951129794,
0.01639600098133087,
-0.0012946130009368062,
-0.0028481485787779093,
0.009452498517930508,
-0.04180688410997391,
0.02587767317891121,
-0.004117234144359827,
-0.013427507132291794,
0.014003701508045197,

-0.019138392060995102,
-0.008781488053500652,
0.03597200661897659,
0.03022465482354164,
-0.01823398657143116,
0.02765730954706669,
0.021647388115525246,
0.025352533906698227,
-0.006615289952605963,
0.010320436209440231,
-0.01785471849143505,
0.009853646159172058,
-0.022610142827033997,
-0.006374601740390062,
0.02189536951482296,
0.020670047029852867,
0.03419237211346626,
0.017256643623113632,
-0.00021641462808474898,
0.01935719884932041,
0.010954978875815868,
-0.031158234924077988,
-0.021122248843312263,
-0.0053170304745435715,
-0.017840132117271423,
0.0030979542061686516,
0.0014842464588582516,
-0.007235245779156685,
0.04819607362151146,
0.037138983607292175,
0.0018471027724444866,
0.03238355740904808,
-0.015345722436904907,
0.0027788595762103796,
0.007512402255088091,
-0.007884375751018524,
0.007782265078276396,
0.010021398775279522,
-0.02349996194243431,
0.008438688702881336,
-0.010590299032628536,
-0.018540317192673683,
0.011954201385378838,
0.011086263693869114,
-0.012479339726269245,
0.017358753830194473,
0.01951765827834606,
0.012151128612458706,

0.014886226505041122,
0.010575711727142334,
-0.0055139572359621525,
-0.013872416689991951,
-0.0017550211632624269,
0.02862006425857544,
0.016439761966466904,
-0.036876413971185684,
0.004142761696130037,
-0.00672104721888998,
-0.03360888361930847,
-0.02597978338599205,
0.02770107239484787,
-0.0025418177247047424,
0.02275601401925087,
0.013587966561317444,
0.018467381596565247,
0.024608587846159935,
0.02609647996723652,
0.020465826615691185,
-0.008037541061639786,
0.0023521841503679752,
0.0007393881096504629,
-0.001940096146427095,
-0.005364438518881798,
-0.015506181865930557,
0.008125063963234425,
-0.030312178656458855,
-0.005149277858436108,
0.004325101617723703,
0.003982302732765675,
0.03028300404548645,
0.007614512462168932,
-0.00472260219976306,
0.01613343134522438,
-0.025848498567938805,
0.026184003800153732,
0.01226782612502575,
0.01661480776965618,
-0.022887298837304115,
0.03877274692058563,
0.022449683398008347,
0.006746574770659208,
0.06284160912036896,
-0.014353794045746326,
0.004624139051884413,
0.021107662469148636,
0.0022136059124022722,

-0.01331080961972475,
0.020670047029852867,
0.004996112082153559,
-0.006527767051011324,
-0.001420427463017404,
-0.014689299277961254,
-0.0077457972802221775,
0.0006181321223266423,
-0.0063089593313634396,
-0.01763591170310974,
-0.0007010967819951475,
0.000883892469573766,
0.09201598912477493,
0.022726839408278465,
-0.000578473205678165,
-0.02088853818178177,
-0.030166305601596832,
0.022522618994116783,
-0.011903146281838417,
0.02059711143374443,
0.018919583410024643,
-0.02808033861219883,
-0.02249344438314438,
-0.0022646610159426928,
-0.023251978680491447,
-0.029086854308843613,
-0.010211031883955002,
0.027992814779281616,
-0.0357094369828701,
0.015141502022743225,
-0.003112541511654854,
-0.008008366450667381,
0.027307217940688133,
0.005988040938973427,
0.01331080961972475,
0.005451961886137724,
0.0021899016574025154,
-0.01861325278878212,
0.02781776897609234,
-0.008489743806421757,
-0.008409514091908932,
-0.01096956618130207,
0.004284986760467291,
0.005492076743394136,
-0.018117288127541542,
-0.01995527371764183,
-0.016162605956196785,
-0.003920306917279959,

```

        0.00014062962145544589,
        -0.010524656623601913,
        -0.006031802389770746,
        0.007282653823494911,
        0.00605368334800005,
        -0.003674148116260767,
        -0.04489936679601669,
        -0.03331713750958443,
        0.00965671893209219,
        -0.0010438957251608372,
        0.013055534102022648,
        -0.018657013773918152,
        -0.05029662698507309
    ]
}
}
]
}

```

```
[18]: from elasticsearch import Elasticsearch
import json

# Initialize Elasticsearch client
es = Elasticsearch(['http://localhost:9200'])

# List of repositories
repositories = [
    ("openai", "openai-cookbook"),
    ("elastic", "elasticsearch"),
    ("langchain-ai", "langchain"),
    ("milvus-io", "pymilvus"),
    ("langchain-ai", "langgraph"),
    ("microsoft", "autogen")
]

# Function to perform semantic search and retrieve top similar issues
def semantic_search(issue_vector, k=5):
    response = es.search(
        index="github_issues",
        knn={
            "field": "GitHub_Issue_vector",
            "query_vector": issue_vector,
            "k": k,
            "num_candidates": 10000
        }
    )

```

```

similar_issues = []
for hit in response["hits"]["hits"]:
    similar_issue = {
        "_score": hit["_score"],
        "_type": hit["_source"]["_type"],
        "_repo": hit["_source"]["_repo"],
        "_issueNumber": hit["_source"]["_issueNumber"],
        "_title": hit["_source"]["_title"],
        "_createdAt": hit["_source"]["_createdAt"],
        "_closedAt": hit["_source"]["_closedAt"],
        "_body": hit["_source"]["_body"]
    }
    similar_issues.append(similar_issue)

return similar_issues

# Iterate through repositories
print("\n" + "-" * 75 + "\n")
for owner, repo in repositories:

    print(f"Top 5 similar issues for {owner}/{repo}:")
    print("\n" + "-" * 75 + "\n")

    # Fetch recent issues for the current repository
    res = es.search(index="github_issues", body={"query": {"match": {"_repo": owner + f"/{repo}"}}})
    hits = res.get('hits', {})

    # Choose an issue from the repository's recent issues
    chosen_issue = hits['hits'][0]['_source'] if hits['hits'] else None

    if chosen_issue:
        issue_vector = chosen_issue.get("GitHub_Issue_vector")
        similar_issues = semantic_search(issue_vector, k=5)

        # Output similar issues
        for similar_issue in similar_issues:
            print(json.dumps(similar_issue, indent=2))
    else:
        print(f"No recent similar issues found for {owner}/"
              f"{repo}")

    print("\n" + "-" * 75 + "\n")

```

Top 5 similar issues for openai/openai-cookbook:

No recent similar issues from last 2 months found for openai/openai-cookbook

Top 5 similar issues for elastic/elasticsearch:

```
{  
  "_score": 1.0,  
  "_type": "issue",  
  "_repo": "elasticsearch",  
  "_issueNumber": "116131",  
  "_title": "[8.16] Resolve pipelines from template if lazy rollover write (#116031)",  
  "_createdAt": "2024-11-02T03:56:02Z",  
  "_closedAt": "2024-11-02T17:12:07Z",  
  "_body": "Backports the following commits to 8.16:\n - Resolve pipelines from template if lazy rollover write (#116031)"  
}  
{  
  "_score": 0.96739626,  
  "_type": "issue",  
  "_repo": "elasticsearch",  
  "_issueNumber": "115367",  
  "_title": "[8.x] Optimize IngestService#resolvePipelinesFromIndexTemplates (#115348)",  
  "_createdAt": "2024-10-22T20:08:04Z",  
  "_closedAt": "2024-10-22T21:17:33Z",  
  "_body": "Backports the following commits to 8.x:\n - Optimize IngestService#resolvePipelinesFromIndexTemplates (#115348)"  
}  
{  
  "_score": 0.96484697,  
  "_type": "issue",  
  "_repo": "elasticsearch",  
  "_issueNumber": "116031",  
  "_title": "Resolve pipeline on lazy rollover write ",  
  "_createdAt": "2024-10-31T17:00:23Z",  
  "_closedAt": "2024-11-02T03:54:55Z",  
  "_body": "This fixes a bug described in  
https://github.com/elastic/elasticsearch/issues/112781 . The issue being that if lazy rollover is set, and a reroute processor always reroutes to another index, the write index of a data stream will never roll over. Because of this, if the
```

```

pipeline is changed in a template, this change will not go into effect. To avoid
this, when lazy rollover is set, we always resolve the pipeline from
templates.\r\n\r\nFixes: https://github.com/elastic/elasticsearch/issues/112781"
}
{
  "_score": 0.960582,
  "_type": "issue",
  "_repo": "elasticsearch",
  "_issueNumber": "115987",
  "_title": "Resolve pipeline on lazy rollover write",
  "_createdAt": "2024-10-30T20:48:33Z",
  "_closedAt": "2024-10-31T20:29:52Z",
  "_body": "If lazy rollover is set, when write with bulk request, always
resolve default (and final) pipeline from index templates.\r\n\r\nRelated to
https://github.com/elastic/elasticsearch/issues/112781"
}
{
  "_score": 0.9534638,
  "_type": "issue",
  "_repo": "elasticsearch",
  "_issueNumber": "115966",
  "_title": "[8.16] [ML] Handle Errors and pre-streaming exceptions (#115868)",
  "_createdAt": "2024-10-30T16:51:34Z",
  "_closedAt": "2024-10-30T17:58:22Z",
  "_body": "Backports the following commits to 8.16:\n - [ML] Handle Errors and
pre-streaming exceptions (#115868)"
}
-----
```

Top 5 similar issues for langchain-ai/langchain:

```

{
  "_score": 1.0,
  "_type": "issue",
  "_repo": "langchain",
  "_issueNumber": "28276",
  "_title": "langchain-chroma== 0.1.4    method get_by_ids is listed in
documentation BUT I am getting NotImplementedError",
  "_createdAt": "2024-11-22T01:13:50Z",
  "_closedAt": "2024-12-31T00:36:30Z",
  "_body": "### Checked other resources\n- [X] I added a very descriptive
title to this issue.\n- [X] I searched the LangChain documentation with the
integrated search.\n- [X] I used the GitHub search to find a similar question
and didn't find it.\n- [X] I am sure that this is a bug in LangChain rather than
my code.\n- [X] The bug is not resolved by updating to the latest stable version
```

```

of LangChain (or the specific integration package).\n\n### Example
Code\n\n```\r\n#\r\n# HuggingFace embedding (no issue)\r\nfrom langchain_huggingface import HuggingFaceEmbeddings\r\nembeddings = HuggingFaceEmbeddings(model=\"sentence-transformers/all-mpnet-base-v2\")\r\n#\r\n# create langchain-chroma persistent client with collection name 'example_collection; (no issue)\r\nfrom langchain_chroma import Chroma\r\nvector_store = Chroma(\r\n    collection_name=\"example_collection\", # collection is \"table\" in vector store\r\n    embedding_function=hf, # hf is huggingface embeddings derived from the previous step\r\n    persist_directory=\"./vectorstore/chroma_langchain_db\", # Where to save data locally, remove if not necessary\r\n    n_documents = [document_1 = Document(\r\n        page_content=\"I had chocolate chip pancakes and scrambled eggs for breakfast this morning.\",\r\n        metadata={\"source\": \"tweet\"},\r\n        id=1),\r\n        document_1,\r\n        n_documents = [\r\n            document_1,\r\n            n_uuids = [str(uuid4()) for _ in range(len(documents))]\r\n            vector_store.add_documents(documents=documents,\r\n            ids=uuids)\r\n            # ERROR ENCONTERED when running get_by_ids attempt to run get_by_ids yields NotImplementedError\r\n            vector_store.get_by_ids(['6314982d-455f-47cc-bf97-6e5324f6af62'])\r\n            # Error Message and Stack Trace (if applicable)\r\n            {\r\n                \"name\": \"NotImplementedError\",\r\n                \"message\": \"Chroma does not yet support get_by_ids.\",\r\n                \"stack\": \"-----\r\n                -----\\r\\nNotImplementedError\r\n                Traceback (most recent call last)\\r\\nCell In[87], line 3\\r\\n        1 # testing get the first two document ids\\r\\n        2 # ids = ['db1e5f74-f18d-4765-a193-d30eaed7552f',\r\n        '12861b34-df54-4e40-8e1e-ae9ea901d378']\\r\\n----> 3 vector_store.get_by_ids(['6314982d-455f-47cc-bf97-6e5324f6af62'])\\r\\n        # get_by_ids() functionality is not available until v0.2.11\\r\\n        # File ~/Documents/0_-_Python_Projects/05_Gen_AI/venv_3_11/lib/python3.11/site-packages/langchain_core/vectorstores/base.py:164, in VectorStore.get_by_ids(self, ids)\r\n        140 \\\\\"\\\\\"\\\\\"Get documents by their IDs.\r\n        141 \\r\\n        142 The returned documents are expected to have the ID field set to the ID of the\r\n        (...)\\r\\n        161 .. versionadded:: 0.2.11\\r\\n        162 \\\\\"\\\\\"\\\\\"\\r\\n        163 msg = f\\\\\"{self.__class__.__name__} does not yet support get_by_ids.\\\\\"\\r\\n--> 164 raise NotImplementedError(msg)\\r\\n\\r\\nNotImplementedError: Chroma does not yet support get_by_ids.\r\n        \\r\\n        \\n### Description\\n        I am just trying to run the vector_store method `get_by_ids` - it is listed as one of the available methods in [here] (https://python.langchain.com/api_reference/chroma/vectorstores/langchain_chroma.vectorstores.Chroma.html)\\r\\n        \\n### System Info\\n        $ python -m langchain_core.sys_info\\r\\n        System Information\\r\\n        OS: Darwin\\r\\n        OS Version: Darwin Kernel Version 23.6.0: Mon Jul 29 21:13:00 PDT 2024; root:xnu-10063.141.2~1/RELEASE_X86_64\\r\\n        Python Version: 3.11.10 (main, Nov 19 2024, 15:24:32) [Clang 12.0.0 (clang-1200.0.32.29)]\\r\\n        Package

```

```

Information\r\n-----\r\n> langchain_core: 0.3.19\r\n> langchain:
0.3.7\r\n> langchain_community: 0.3.4\r\n> langsmith: 0.1.143\r\n>
langchain_chroma: 0.1.4\r\n> langchain_experimental: 0.3.3\r\n> langchain_groq:
0.2.1\r\n> langchain_huggingface: 0.1.2\r\n> langchain_text_splitters:
0.3.2\r\n\r\nOptional packages not
installed\r\n-----\r\n> langgraph\r\n>
langserve\r\n\r\nOther Dependencies\r\n-----\r\n> aiohttp:
3.11.6\r\n> async-timeout: Installed. No version info available.\r\n> chromadb:
0.5.20\r\n> dataclasses-json: 0.6.7\r\n> fastapi: 0.115.5\r\n> groq: 0.12.0\r\n>
httpx: 0.27.2\r\n> httpx-sse: 0.4.0\r\n> huggingface-hub: 0.26.2\r\n> jsonpatch:
1.33\r\n> numpy: 1.26.4\r\n> orjson: 3.10.11\r\n> packaging: 24.2\r\n> pydantic:
2.9.2\r\n> pydantic-settings: 2.6.1\r\n> PyYAML: 6.0.2\r\n> requests:
2.32.3\r\n> requests-toolbelt: 1.0.0\r\n> sentence-transformers: 3.3.1\r\n>
SQLAlchemy: 2.0.36\r\n> tenacity: 9.0.0\r\n> tokenizers: 0.20.3\r\n>
transformers: 4.46.3\r\n> typing-extensions: 4.12.2"
}
{
    "_score": 0.9405959,
    "_type": "issue",
    "_repo": "langchain",
    "_issueNumber": "28256",
    "_title": "DOC: Docs describe FAISS get_by_id but it isn't implemented",
    "_createdAt": "2024-11-21T10:41:14Z",
    "_closedAt": "2024-12-31T00:36:30Z",
    "_body": "### URL\r\nhttps://python.langchain.com/api_reference/community/vectorstores/langchain_community.vectorstores.faiss.FAISS.html#langchain_community.vectorstores.faiss.FAISS.get_by_ids\r\n### Checklist\r\n[X] I added a very descriptive title to this issue.\r\n[X] I included a link to the documentation page I am referring to (if applicable).\r\n### Issue with current documentation:\r\nDocs for FAISS describe get_by_id but it doesn't seem to be implemented as yet.\r\n### Idea or request for content:\r\nI suggest either implementing get_by_id or removing it from the docs."
}
{
    "_score": 0.9356828,
    "_type": "issue",
    "_repo": "langchain",
    "_issueNumber": "27876",
    "_title": "RedisVectorStore add_texts uses wrong function parameter for ids",
    "_createdAt": "2024-11-04T09:33:16Z",
    "_closedAt": "2024-12-31T00:36:30Z",
    "_body": "### Checked other resources\r\n[X] I added a very descriptive title to this issue.\r\n[X] I searched the LangChain documentation with the integrated search.\r\n[X] I used the GitHub search to find a similar question and didn't find it.\r\n[X] I am sure that this is a bug in LangChain rather than my code.\r\n[X] The bug is not resolved by updating to the latest stable version of LangChain (or the specific integration package).\r\n### Example Code\r\nThe RedisVectorStore implementation uses a different name for ids, namely 'keys'. In

```

```

all other vector stores and implementations it is 'ids'.\r\n```\python\r\n
vector_store = RedisVectorStore(EMBEDDING_MODEL, config=config)\r\n      doc =
Document(\r\n          \"Some text\", \r\n          id=\"unique_id1\"\r\n
)\r\n      added_ids = vector_store.add_documents([doc]) # does not work, ids
are not used!\r\n      added_ids2 = vector_store.add_documents([doc],
keys=[doc.id]) # works, since RedisVectorStore is looking for 'keys' instead of
'ids'\r\n```\n### Error Message and Stack Trace (if applicable)\n\nNo
response_\n### Description\nThe RedisVectorStore implementation uses a
different name for ids, namely 'keys'. In all other vector stores and
implementations it is 'ids'. This seems wrong and it's only possible to set your
own ids if you inspect the code and add 'keys' instead\n\n### System
Info\nSystem Information\r\n-----\r\n> OS: Darwin\r\n> OS
Version: Darwin Kernel Version 23.6.0: Wed Jul 31 20:49:46 PDT 2024;
root:xnu-10063.141.1.700.5~1/RELEASE_ARM64_T8103\r\n> Python Version: 3.11.9
(main, Apr 2 2024, 08:25:04) [Clang 15.0.0 (clang-1500.3.9.4)]\r\n\r\nPackage
Information\r\n-----\r\n> langchain_core: 0.3.14\r\n> langchain:
0.3.6\r\n> langchain_community: 0.3.4\r\n> langsmith: 0.1.137\r\n>
langchain_redis: 0.1.1\r\n> langchain_text_splitters: 0.3.1\r\n\r\nOptional
packages not installed\r\n-----\r\n> langgraph\r\n>
langserve\r\n\r\nOther Dependencies\r\n-----\r\n> aiohttp:
3.10.10\r\n> async-timeout: Installed. No version info available.\r\n>
dataclasses-json: 0.6.7\r\n> httpx: 0.27.2\r\n> httpx-sse: 0.4.0\r\n> jsonpatch:
1.33\r\n> numpy: 1.26.4\r\n> orjson: 3.10.10\r\n> packaging: 24.1\r\n> pydantic:
2.9.2\r\n> pydantic-settings: 2.6.0\r\n> python-ulid: 2.7.0\r\n> PyYAML:
6.0.2\r\n> redisvl: 0.3.5\r\n> requests: 2.32.3\r\n> requests-toolbelt:
1.0.0\r\n> SQLAlchemy: 2.0.36\r\n> tenacity: 8.5.0\r\n> typing-extensions:
4.12.2"
}
{
    "_score": 0.9345304,
    "_type": "issue",
    "_repo": "langchain",
    "_issueNumber": "26860",
    "_title": "Choma similarity_search_with_score (and similar methods) don't
populate the document.id property of the returned documents.",
    "_createdAt": "2024-09-25T16:03:13Z",
    "_closedAt": "2024-10-24T21:47:37Z",
    "_body": "### Checked other resources\n[X] I added a very descriptive
title to this issue.\n[X] I searched the LangChain documentation with the
integrated search.\n[X] I used the GitHub search to find a similar question
and didn't find it.\n[X] I am sure that this is a bug in LangChain rather than
my code.\n[X] The bug is not resolved by updating to the latest stable version
of LangChain (or the specific integration package).\n\n### Example Code\n\nI
want to get the document id from the documents I find with
`similarity_search_with_score` but the id's aren't being set to the documents
when they are created.\r\n\r\nHere is my test code:\r\n```python\r\n\ndef
test_basic_operations_with_langchain():\r\n    token =
os.environ.get('CHROMA_TOKEN')\r\n    client = chromadb.HttpClient(\r\n        s

```

```

ettings=Settings(chroma_client_auth_provider=\"chromadb.auth.token_authn.TokenAu
thClientProvider\",\r\n
chroma_client_auth_credentials=token))\r\n      test_collection_name =
\"test_collection\"\r\n\r\n      try:\r\n
client.delete_collection(test_collection_name)\r\n      except: # noqa: E722\r\n
pass\r\n\r\n      collection =
client.get_or_create_collection(test_collection_name)\r\n      assert
collection.name == test_collection_name\r\n      assert collection.count() ==
0\r\n\r\n      EMBEDDING_MODEL_NAME = \"sentence-transformers/all-mpnet-
base-v2\"\r\n      EMBEDDING_MODEL_CHUNK_SIZE = 384\r\n\r\n      embedding_model =
HuggingFaceEmbeddings(\r\n          model_name=EMBEDDING_MODEL_NAME,\r\n          # multi_process=True, # I found this causes crashes and slowness\r\n
model_kwargs={"device": "cuda" if torch.cuda.is_available() else
"cpu"},\r\n          encode_kwargs={\r\n              "normalize_embeddings": True\r\n          }, # set True for cosine similarity\r\n      )\r\n      vector_db =
Chroma(\r\n          embedding_function=embedding_model,\r\n          client=client,\r\n          collection_name=test_collection_name,\r\n          collection_metadata={"hnsw:space": "cosine"},\r\n          # Make new
uid\r\n          id = str(uuid.uuid4())\r\n          test_doc = Document(\"hello
world\")\r\n          test_doc.id = id\r\n          test_doc.metadata[\"test\"] =
\"test\"\r\n          vector_db.add_documents([test_doc])\r\n          assert
collection.count() == 1\r\n\r\n      docs =
vector_db.similarity_search_with_score(\"hello world\", 1)\r\n      assert
len(docs) == 1\r\n      assert docs[0][0].page_content == \"hello world\"\r\n
assert docs[0][0].id == id\r\n\r\n      client.delete_collection(test_collection_name)\r\n      ````\r\n      # Error Message
and Stack Trace (if applicable)\r\nTest
output:\r\n      ````\r\n      ======\r\n      FAILURES
=====````\r\n-----\r\n      test_basic_operations_with_langchain ..... \r\n      def
test_basic_operations_with_langchain():\r\n          token =
os.environ.get('CHROMA_TOKEN')\r\n          client = chromadb.HttpClient(\r\n
settings=Settings(chroma_client_auth_provider=\"chromadb.auth.token_authn.TokenA
uthClientProvider\"),\r\n
chroma_client_auth_credentials=token))\r\n          test_collection_name =
\"test_collection\"\r\n          try:\r\n
client.delete_collection(test_collection_name)\r\n          except: # noqa:
E722\r\n          pass\r\n          collection =
client.get_or_create_collection(test_collection_name)\r\n          assert
collection.name == test_collection_name\r\n          assert collection.count() ==
0\r\n          EMBEDDING_MODEL_NAME = \"sentence-transformers/all-mpnet-
base-v2\"\r\n          EMBEDDING_MODEL_CHUNK_SIZE = 384\r\n          embedding_model =
HuggingFaceEmbeddings(\r\n              model_name=EMBEDDING_MODEL_NAME,\r\n              # multi_process=True, # I
found this causes crashes and slowness\r\n              model_kwargs={"device": "cuda" if torch.cuda.is_available() else
"cpu"},\r\n              encode_kwargs={\r\n                  "normalize_embeddings": True\r\n              }, # set True for cosine similarity\r\n          )\r\n          vector_db =

```

```

Chroma(\r\n
    embedding_function=embedding_model,\r\n
    client=client,\r\n
        collection_name=test_collection_name,\r\n
    collection_metadata={"hnsw:space": "cosine"},\r\n
) \r\n
# Make
new uid\r\n
    id = str(uuid.uuid4())\r\n
    test_doc = Document("hello
world")\r\n
    test_doc.id = id\r\n
    test_doc.metadata["test"] =
"test"\r\n
    vector_db.add_documents([test_doc])\r\n
    assert
collection.count() == 1\r\n
    docs =
vector_db.similarity_search_with_score("hello world", 1)\r\n
    assert
len(docs) == 1\r\n
    assert docs[0][0].page_content == "hello world"\r\n
assert docs[0][0].id == id\r\nE
AssertionError: assert None ==
'808971a7-f042-4306-864b-97af29c4837d'\r\nE
+ where None =
Document(metadata={'test': 'test'}, page_content='hello
world').id\r\nr\nncode/tests/test_basic_chroma_setup.py:99:
AssertionError\r\n=====
===== warnings summary
=====\\r\\n<frozen importlib._bootstrap>:488\\r\\n
<frozen importlib._bootstrap>:488: DeprecationWarning: Type
google._upb._message.MessageMapContainer uses PyType"
}
{
    "_score": 0.93122804,
    "_type": "issue",
    "_repo": "langchain",
    "_issueNumber": "27995",
    "_title": "langchain_chroma: added document.id support",
    "_createdAt": "2024-11-08T18:15:12Z",
    "_closedAt": "2024-12-31T00:36:30Z",
    "_body": "Description:\\r\\n* Added internal `Document.id` support to Chroma
VectorStore\\r\\n\\r\\nDependencies:\\r\\n* https://github.com/langchain-
ai/langchain/pull/27968 should be merged first and this PR should be re-based on
top of those changes.\\r\\n\\r\\nTests:\\r\\n* Modified/Added tests for `Document.id`
support. All tests are passing.\\r\\n\\r\\n\\r\\nNote: I am not a member of the Chroma
team."
}
-----
```

Top 5 similar issues for milvus-io/pymilvus:

No recent similar issues from last 2 months found for milvus-io/pymilvus

Top 5 similar issues for langchain-ai/langgraph:

```
{
  "_score": 1.0,
  "_type": "issue",
  "_repo": "langchain",
  "_issueNumber": "28276",
  "_title": "langchain-chroma== 0.1.4 method get_by_ids is listed in documentation BUT I am getting NotImplementedError",
  "_createdAt": "2024-11-22T01:13:50Z",
  "_closedAt": "2024-12-31T00:36:30Z",
  "_body": "### Checked other resources\n\n- [X] I added a very descriptive title to this issue.\n- [X] I searched the LangChain documentation with the integrated search.\n- [X] I used the GitHub search to find a similar question and didn't find it.\n- [X] I am sure that this is a bug in LangChain rather than my code.\n- [X] The bug is not resolved by updating to the latest stable version of LangChain (or the specific integration package).\n\n### Example Code\n```\r\n# HuggingFace embedding (no issue)\r\nfrom langchain_huggingface import HuggingFaceEmbeddings\r\nembeddings = HuggingFaceEmbeddings(model=\"sentence-transformers/all-mpnet-base-v2\")\r\n# create langchain-chroma persistent client with collection name 'example_collection' (no issue)\r\nfrom langchain_chroma import Chroma\r\nvector_store = Chroma(\r\n    collection_name=\"example_collection\", # collection is \"table\" in vector store\r\n    embedding_function=hf, # hf is huggingface embeddings derived from the previous step\r\n    persist_directory=\"./vectorstore/chroma_langchain_db\", # Where to save data locally, remove if not necessary\r\n    metadata={\"source\": \"tweet\"},\r\n    id=1,\r\n)\r\n# documents = [\r\n    document_1,\r\n]\r\n# uuids = [str(uuid4()) for _ in range(len(documents))]\r\nvector_store.add_documents(documents=documents,\r\n    ids=uuids)\r\n# ERROR ENCONTERED when running get_by_ids\r\n# attempt to run get_by_ids yields NotImplementedError\r\nvector_store.get_by_ids(['6314982d-455f-47cc-bf97-6e5324f6af62'])\r\n```\n### Error Message and Stack Trace (if applicable)\n{\n    "name": "NotImplementedError",\n    "message": "Chroma does not yet support get_by_ids.",\n    "stack": "\r\n-----\r\nTraceback (most recent call last)\r\nCell In[87], line 3\r\n      1 # testing get the first two document ids\r\n      2 # ids =\r\n['db1e5f74-f18d-4765-a193-d30eaed7552f',\r\n '12861b34-df54-4e40-8e1e-ae9ea901d378']\r\n      3\r\nvector_store.get_by_ids(['6314982d-455f-47cc-bf97-6e5324f6af62'])\r\n      5 #\r\nget_by_ids() functionality is not available until v0.2.11\r\nFile ~/Documents/0_-_Python_Projects/05_Gen_AI/venv_3_11/lib/python3.11/site-packages/langchain_core/vectorstores/base.py:164, in
```

```

VectorStore.get_by_ids(self, ids)\r\n    140 \\"\\\"\\\"\\\"Get documents by their
IDs.\r\n    141 \r\n    142 The returned documents are expected to have the ID
field set to the ID of the\r\n    (...)\\r\\n    161 .. versionadded:: 0.2.11\\r\\n
162 \\"\\\"\\\"\\\"\\\"\\r\\n    163 msg = f\\\"\\\"{self.__class__.__name__} does not yet
support get_by_ids.\\\"\\\"\\r\\n--> 164 raise
NotImplementedError(msg)\r\n\r\nNotImplementedError: Chroma does not yet support
get_by_ids.\\"\\r\\n}\\n\\n### Description\\n\\nI am just trying to run the
vector_store method `get_by_ids` - it is listed as one of the available methods
in [here] (https://python.langchain.com/api\_reference/chroma/vectorstores/langchain\_chroma.vectorstores.Chroma.html)\\r\\n\\n### System Info\\n\\n$ python -m
langchain_core.sys_info\\r\\n\\r\\nSystem Information\\r\\n-----\\r\\n> OS:
Darwin\\r\\n> OS Version: Darwin Kernel Version 23.6.0: Mon Jul 29 21:13:00 PDT
2024; root:xnu-10063.141.2~1/RELEASE_X86_64\\r\\n> Python Version: 3.11.10 (main,
Nov 19 2024, 15:24:32) [Clang 12.0.0 (clang-1200.0.32.29)]\\r\\n\\r\\nPackage
Information\\r\\n-----\\r\\n> langchain_core: 0.3.19\\r\\n> langchain:
0.3.7\\r\\n> langchain_community: 0.3.4\\r\\n> langsmith: 0.1.143\\r\\n>
langchain_chroma: 0.1.4\\r\\n> langchain_experimental: 0.3.3\\r\\n> langchain_groq:
0.2.1\\r\\n> langchain_huggingface: 0.1.2\\r\\n> langchain_text_splitters:
0.3.2\\r\\n\\r\\nOptional packages not
installed\\r\\n-----\\r\\n> langgraph\\r\\n>
langserve\\r\\n\\r\\nOther Dependencies\\r\\n-----\\r\\n> aiohttp:
3.11.6\\r\\n> async-timeout: Installed. No version info available.\\r\\n> chromadb:
0.5.20\\r\\n> dataclasses-json: 0.6.7\\r\\n> fastapi: 0.115.5\\r\\n> groq: 0.12.0\\r\\n>
httpx: 0.27.2\\r\\n> httpx-sse: 0.4.0\\r\\n> huggingface-hub: 0.26.2\\r\\n> jsonpatch:
1.33\\r\\n> numpy: 1.26.4\\r\\n> orjson: 3.10.11\\r\\n> packaging: 24.2\\r\\n> pydantic:
2.9.2\\r\\n> pydantic-settings: 2.6.1\\r\\n> PyYAML: 6.0.2\\r\\n> requests:
2.32.3\\r\\n> requests-toolbelt: 1.0.0\\r\\n> sentence-transformers: 3.3.1\\r\\n>
SQLAlchemy: 2.0.36\\r\\n> tenacity: 9.0.0\\r\\n> tokenizers: 0.20.3\\r\\n>
transformers: 4.46.3\\r\\n> typing-extensions: 4.12.2"
}
{
    "_score": 0.9405959,
    "_type": "issue",
    "_repo": "langchain",
    "_issueNumber": "28256",
    "_title": "DOC: Docs describe FAISS get_by_id but it isn't implemented",
    "_createdAt": "2024-11-21T10:41:14Z",
    "_closedAt": "2024-12-31T00:36:30Z",
    "_body": "### URL\\nhttps://python.langchain.com/api\_reference/community/vectorstores/langchain\_community.vectorstores.faiss.FAISS.html#langchain\_community.vectorstores.faiss.FAISS.get\_by\_ids\\n\\n### Checklist\\n\\n- [X] I added a very descriptive title to this issue.\\n- [X] I included a link to the documentation page I am referring to (if applicable).\\n\\n### Issue with current documentation:\\n\\nDocs for FAISS describe get_by_id but it doesn't seem to be implemented as yet. \\n\\n### Idea or request for content:\\n\\nI suggest either implementing get_by_id or removing it from the docs."
}
{

```

```

"_score": 0.9356828,
"_type": "issue",
"_repo": "langchain",
"_issueNumber": "27876",
"_title": "RedisVectorStore add_texts uses wrong function parameter for ids",
"_createdAt": "2024-11-04T09:33:16Z",
"_closedAt": "2024-12-31T00:36:30Z",
"_body": "### Checked other resources\n\n- [X] I added a very descriptive title to this issue.\n- [X] I searched the LangChain documentation with the integrated search.\n- [X] I used the GitHub search to find a similar question and didn't find it.\n- [X] I am sure that this is a bug in LangChain rather than my code.\n- [X] The bug is not resolved by updating to the latest stable version of LangChain (or the specific integration package).\n\n### Example Code\n\nThe RedisVectorStore implementation uses a different name for ids, namely 'keys'. In all other vector stores and implementations it is 'ids'.\r\n```\r\nvector_store = RedisVectorStore(EMBEDDING_MODEL, config=config)\r\n\r\ndoc = Document(\r\n    \"Some text\", \r\n    id=\"unique_id1\"\r\n)\r\n\r\n    added_ids = vector_store.add_documents([doc]) # does not work, ids are not used!\r\n    added_ids2 = vector_store.add_documents([doc],\r\n        keys=[doc.id]) # works, since RedisVectorStore is looking for 'keys' instead of 'ids'\r\n```\r\n### Error Message and Stack Trace (if applicable)\r\nNo response_\r\n\n### Description\n\nThe RedisVectorStore implementation uses a different name for ids, namely 'keys'. In all other vector stores and implementations it is 'ids'. This seems wrong and it's only possible to set your own ids if you inspect the code and add 'keys' instead.\r\n\n### System Info\r\nSystem Information\r\n-----\r\nOS: Darwin\r\nOS Version: Darwin Kernel Version 23.6.0: Wed Jul 31 20:49:46 PDT 2024;\r\nroot:xnu-10063.141.1.700.5~1/RELEASE_ARM64_T8103\r\nPython Version: 3.11.9 (main, Apr 2 2024, 08:25:04) [Clang 15.0.0 (clang-1500.3.9.4)]\r\n\r\nPackage Information\r\n-----\r\nlangchain_core: 0.3.14\r\nlangchain: 0.3.6\r\nlangchain_community: 0.3.4\r\nlangsmith: 0.1.137\r\nlangchain_redis: 0.1.1\r\nlangchain_text_splitters: 0.3.1\r\nOptional packages not installed\r\n-----\r\nlanggraph\r\nlangserve\r\nOther Dependencies\r\n-----\r\naiohttp: 3.10.10\r\nasync-timeout: Installed. No version info available.\r\ndataclasses-json: 0.6.7\r\nhttpx: 0.27.2\r\nhttpx-sse: 0.4.0\r\njsonpatch: 1.33\r\nnumpy: 1.26.4\r\norjson: 3.10.10\r\npackaging: 24.1\r\npydantic: 2.9.2\r\npydantic-settings: 2.6.0\r\npython-ulid: 2.7.0\r\nPyYAML: 6.0.2\r\nredisvl: 0.3.5\r\nrequests: 2.32.3\r\nrequests-toolbelt: 1.0.0\r\nSQLAlchemy: 2.0.36\r\ntenacity: 8.5.0\r\ntyping-extensions: 4.12.2"
}
{
    "_score": 0.9345304,
    "_type": "issue",
    "_repo": "langchain",
    "_issueNumber": "26860",
    "_title": "Choma similarity_search_with_score (and similar methods) don't"
}

```

populate the document.id property of the returned documents.",
 "_createdAt": "2024-09-25T16:03:13Z",
 "_closedAt": "2024-10-24T21:47:37Z",
 "_body": "### Checked other resources\n- [X] I added a very descriptive title to this issue.\n- [X] I searched the LangChain documentation with the integrated search.\n- [X] I used the GitHub search to find a similar question and didn't find it.\n- [X] I am sure that this is a bug in LangChain rather than my code.\n- [X] The bug is not resolved by updating to the latest stable version of LangChain (or the specific integration package).\n\n### Example Code\n\nI want to get the document id from the documents I find with `similarity_search_with_score` but the id's aren't being set to the documents when they are created.\nHere is my test code:\n```python\ndef test_basic_operations_with_langchain():\n token = os.environ.get('CHROMA_TOKEN')\n client = chromadb.HttpClient(\n settings=Settings(chroma_client_auth_provider=\"chromadb.auth.token_authn.TokenAuthClientProvider\"),\n chroma_client_auth_credentials=token)\n test_collection_name = \"test_collection\"\n try:\n client.delete_collection(test_collection_name)\n except: # noqa: E722\n pass\n collection = client.get_or_create_collection(test_collection_name)\n assert collection.name == test_collection_name\n assert collection.count() == 0\n EMBEDDING_MODEL_NAME = \"sentence-transformers/all-mpnet-base-v2\"\n EMBEDDING_MODEL_CHUNK_SIZE = 384\n embedding_model = HuggingFaceEmbeddings(\n model_name=EMBEDDING_MODEL_NAME,\n multi_process=True, # I found this causes crashes and slowness\n model_kwargs={"device": \"cuda\" if torch.cuda.is_available() else \"cpu\"},\n encode_kwargs={"normalize_embeddings": True}\n)\n # set True for cosine similarity\n vector_db = Chroma(\n embedding_function=embedding_model,\n client=client,\n collection_name=test_collection_name,\n collection_metadata={"hnsw:space": \"cosine"},\n # Make new uid\n id = str(uuid.uuid4())\n)\n test_doc = Document(\"hello world\")\n test_doc.id = id\n test_doc.metadata[\"test\"] = \"test\"\n vector_db.add_documents([test_doc])\n assert collection.count() == 1\n docs = vector_db.similarity_search_with_score(\"hello world\", 1)\n assert len(docs) == 1\n assert docs[0][0].page_content == \"hello world\"\n assert docs[0][0].id == id\n\n client.delete_collection(test_collection_name)\n\n### Error Message and Stack Trace (if applicable)\nTest output:\n```===== FAILURES ======\n\n=====\n\ntest_basic_operations_with_langchain _____ \n\ndef test_basic_operations_with_langchain():\n token = os.environ.get('CHROMA_TOKEN')\n client = chromadb.HttpClient(\n settings=Settings(chroma_client_auth_provider=\"chromadb.auth.token_authn.TokenAuthClientProvider\"),

```

chroma_client_auth_credentials=token))\r\n      test_collection_name =
\"test_collection\"\r\n      \r\n      try:\r\n        client.delete_collection(test_collection_name)\r\n      except: # noqa:
E722\r\n          pass\r\n          \r\n          collection =
client.get_or_create_collection(test_collection_name)\r\n      assert
collection.name == test_collection_name\r\n      assert collection.count() ==
0\r\n      EMBEDDING_MODEL_NAME = \"sentence-transformers/all-mpnet-
base-v2\"\r\n      EMBEDDING_MODEL_CHUNK_SIZE = 384\r\n      \r\nembedding_model = HuggingFaceEmbeddings(\r\nmodel_name=EMBEDDING_MODEL_NAME,\r\n      # multi_process=True, # I
found this causes crashes and slowness\r\n      model_kwargs={"device\":
\"cuda\" if torch.cuda.is_available() else \"cpu\"},\r\nencode_kwargs={\r\n      \"normalize_embeddings\": True\r\n},
# set True for cosine similarity\r\n      )\r\n      vector_db =
Chroma(\r\n      embedding_function=embedding_model,\r\nclient=client,\r\n      collection_name=test_collection_name,\r\n      collection_metadata={"hnsw:space": \"cosine"},\r\n      )\r\n      # Make
new uid\r\n      id = str(uuid.uuid4())\r\n      test_doc = Document(\"hello
world\")\r\n      test_doc.id = id\r\n      test_doc.metadata[\"test\"] =
\"test\"\r\n      vector_db.add_documents([test_doc])\r\n      assert
collection.count() == 1\r\n      \r\n      docs =
vector_db.similarity_search_with_score(\"hello world\", 1)\r\n      assert
len(docs) == 1\r\n      assert docs[0][0].page_content == \"hello world\"\r\nassert docs[0][0].id == id\r\nE AssertionError: assert None ==
'808971a7-f042-4306-864b-97af29c4837d'\r\nE + where None =
Document(metadata={'test': 'test'}, page_content='hello
world').id\r\nr\ncode/tests/test_basic_chroma_setup.py:99:
AssertionError\r\n===== warnings summary
=====\\r\\n<frozen importlib._bootstrap>:488\\r\\n
<frozen importlib._bootstrap>:488: DeprecationWarning: Type
google._upb._message.MessageMapContainer uses PyType"
}
{
  "_score": 0.93122804,
  "_type": "issue",
  "_repo": "langchain",
  "_issueNumber": "27995",
  "_title": "langchain_chroma: added document.id support",
  "_createdAt": "2024-11-08T18:15:12Z",
  "_closedAt": "2024-12-31T00:36:30Z",
  "_body": "Description:\\r\\n* Added internal `Document.id` support to Chroma
VectorStore\\r\\n\\r\\nDependencies:\\r\\n* https://github.com/langchain-
ai/langchain/pull/27968 should be merged first and this PR should be re-based on
top of those changes.\\r\\n\\r\\nTests:\\r\\n* Modified/Added tests for `Document.id` 
support. All tests are passing.\\r\\n\\r\\nNote: I am not a member of the Chroma
team."
}

```

Top 5 similar issues for microsoft/autogen:

No recent similar issues from last 2 months found for microsoft/autogen

[]: