# **Anomaly Detection Using Vector Search**

Anomaly detection refers to identifying patterns or instances that deviate significantly from the norm within a dataset. It helps detect unusual behavior or outliers.

Vectors and language models (LLMs) can aid in querying and discovering anomalies by:

* Representing data points as vectors, allowing efficient similarity comparisons.
* Leveraging LLMs to understand context and identify unexpected patterns in textual or sequential data.

For this, we need a dataset with anomalies. Why not use a dataset that auto-updates live so that constant anomalies occur, which can then be automatically detected and presented in a readable format for the user to analyze and apply in the real world?

I will be using a lot of references to articles and assets present in the links below.

For this tutorial, I will be using [1].

<https://money.rediff.com/news>

And something special for you guys: Llama 3, which was just released a few days ago.

This Lllama 3-8B-Instruct can be set up using [2] and can be downloaded from [3].

https://medium.com/@AyushmanPranav/how-to-install-and-run-llama2-locally-on-windows-for-free-05bd5032c6e3?sk=8b8b54ccafd3f828bde4ab97c1913a28

https://huggingface.co/MaziyarPanahi/Meta-Llama-3-8B-Instruct-GGUF/tree/main

## **Code**

<https://github.com/heathbrew/Vector-Databases-can-help-with-Anomaly-Detection>

Git clone this repo [2] and follow along for the set-up.

## **Indexing Web Content to Create a Dataset**

Let's explore all the available content.

import requests

from bs4 import BeautifulSoup

import pandas as pd

# URL for Rediff Money news

url = "https://money.rediff.com/news"

# Fetch HTML content using requests

response = requests.get(url)

if response.status\_code == 200:

# Parse HTML content using BeautifulSoup

soup = BeautifulSoup(response.content, "html.parser")

soup

This will parse the HTML for the provided URL. In a production application, you can use an array of URLs (or a database of URLs) that you index regularly to keep fresh data in your vector store.

Now, let’s convert this to a usable format.

import requests

from bs4 import BeautifulSoup

import pandas as pd

# URL for Rediff Money news

url = "https://money.rediff.com/news"

# Fetch HTML content using requests

response = requests.get(url)

if response.status\_code == 200:

# Parse HTML content using BeautifulSoup

soup = BeautifulSoup(response.content, "html.parser")

# Find all news items

news\_items = soup.find\_all("div", class\_="rtnews\_row\_more")

# Extract relevant information from each news item

news\_data = []

for item in news\_items:

title = item.find("p").text.strip()

link = item.find("a")["href"]

summary = item.find("div").text.strip()

published = item.find("span", class\_="timeago").text.strip()

news\_data.append({

"title": title,

"summary": summary,

"link": link,

"published": published

})

# Create pandas DataFrame

df = pd.DataFrame(news\_data)

# Save DataFrame to CSV

df.to\_csv("Dataset/financial\_news.csv", index=False)

print("Financial news scraped and saved to financial\_news.csv")

else:

print("Failed to fetch the webpage")

Now, I have the financial news indexed and saved to financial\_news.csv.

Let’s view it.

import pandas as pd

# Read the CSV file into a DataFrame

df = pd.read\_csv("Dataset/financial\_news.csv")

# Display the first few rows of the DataFrame

df.head()

As you can see, using a live dataset instead of a pre-made one is much better because it allows the user to track the latest financial news and discover events that can positively or negatively impact a stock they are interested in. This can enable the user to make buy and sell decisions regarding that stock.

for column in df.columns:

print(f"{column}: {df[column][0]}")

## **Creating a Qdrant Database**

In the last project [4], I used [Qdrant](https://qdrant.tech/) which is a locally supported vector store for RAG.

<https://medium.com/devops-dev/steps-to-monitoring-dspy-qdrant-powered-rag-with-prometheus-or-grafana-b642335cbd50>

For this project, I will once again use Qdrant, but this time for financial data, to create a vector store.

Docker Desktop must be installed.

In [5] I have included a file named Qdrant.ps1.

You can use this to pull the Qdrant image and then run it on port 6333. You will see the local UI, which displays the vector stores.

#Use this code to create a qdrant collection  
from qdrant\_client import QdrantClient

from qdrant\_client.http.models import Distance, VectorParams

# Initialize Qdrant client

qdrant\_client = QdrantClient(host='localhost', port=6333)

collection\_name = "Finance Outliers"

# Specify the vectors' configuration

vectors\_config = VectorParams(

size=model.config.hidden\_size, # The size of your embeddings

distance=Distance.COSINE # The distance metric for the vector space

)

# Create or recreate the collection with the specified configuration

qdrant\_client.recreate\_collection(

collection\_name=collection\_name,

vectors\_config=vectors\_config,

# Optionally, you can specify other parameters for the collection

)

## **Feeding the Dataset to Vector Store**

Take a look at the general dataset:

import pandas as pd

# Load the CSV file into a pandas DataFrame

df = pd.read\_csv("Dataset/financial\_news.csv")

df.head()

### 

### **Loading the Embedding Model**

I will be using the MINI-LM12-V2 [4] to embed the news.

from sentence\_transformers import SentenceTransformer

from transformers import AutoTokenizer, AutoModel

from pathlib import Path

def download\_model\_and\_tokenizer(model\_name, save\_path):

"""

Download and save both the model and the tokenizer to the specified directory.

Parameters:

model\_name (str): Name of the model to download.

save\_path (str or Path): Path to the directory where the model and tokenizer will be saved.

"""

# Create the save path if it doesn't exist

save\_path = Path(save\_path)

save\_path.mkdir(parents=True, exist\_ok=True)

# Initialize tokenizer and model

tokenizer = AutoTokenizer.from\_pretrained(model\_name)

model = AutoModel.from\_pretrained(model\_name)

# Save tokenizer

tokenizer.save\_pretrained(save\_path)

# Save model

model.save\_pretrained(save\_path)

# Example usage

model\_name = 'sentence-transformers/all-MiniLM-L12-v2' # Model name to download

save\_path = Path("MiniLM-L12-v2/") # Path where model and tokenizer will be saved

download\_model\_and\_tokenizer(model\_name, save\_path)

Load the model and the tokenizer.

from transformers import AutoTokenizer, AutoModel

def load\_model\_and\_tokenizer(model\_path):

"""

Load the model and tokenizer from the specified directory.

Parameters:

model\_path (str or Path): Path to the directory containing the saved model and tokenizer.

Returns:

tokenizer (transformers.PreTrainedTokenizer): Loaded tokenizer.

model (transformers.PreTrainedModel): Loaded model.

"""

model\_path = Path(model\_path)

tokenizer = AutoTokenizer.from\_pretrained(model\_path)

model = AutoModel.from\_pretrained(model\_path)

return tokenizer, model

# Load the model and tokenizer

model\_path = Path("MiniLM-L12-v2/")

tokenizer, model = load\_model\_and\_tokenizer(model\_path)

Merge the title and the summary.

df['news'] = df.apply(lambda row: row['title'] + ' ' + row['summary'], axis=1)

df.head()

Embed the news.

import torch

#Mean Pooling - Take attention mask into account for correct averaging

def mean\_pooling(model\_output, attention\_mask):

token\_embeddings = model\_output[0] #First element of model\_output contains all token embeddings

input\_mask\_expanded = attention\_mask.unsqueeze(-1).expand(token\_embeddings.size()).float()

return torch.sum(token\_embeddings \* input\_mask\_expanded, 1) / torch.clamp(input\_mask\_expanded.sum(1), min=1e-9)

def generate\_embedding(text):

# Tokenize input text

encoded\_input = tokenizer(text, padding=True, truncation=True, return\_tensors='pt')

# Compute token embeddings with model

with torch.no\_grad():

model\_output = model(\*\*encoded\_input)

# Perform mean pooling

sentence\_embedding = mean\_pooling(model\_output, encoded\_input['attention\_mask'])

# Convert to numpy for FAISS compatibility and ensure it's 2D

return sentence\_embedding.cpu().numpy().reshape(1, -1)

# Generate embeddings for the 'input' column

df['encoded\_news'] = df['news'].apply(lambda x: generate\_embedding(x)[0].tolist())

df.head()

## **Pushing the Data to Qdrant Vector Store**

### **Querying Qdrant**

Query the anomalies in any way you want.

from langchain.vectorstores import Qdrant

url = "http://localhost:6333" # URL where the Qdrant service is running

collection\_name = "Finance Outliers" # Name of the collection in Qdrant

# Initialize the Qdrant client with the specified URL

client = QdrantClient(

url=url,

prefer\_grpc=False # Indicates whether to use gRPC for communication

)

logging.info(f"QdrantClient initialized: {client}") # Prints the client information

logging.info(f"#################################") # Prints a separator line

# Create a Qdrant object with the specified client, embeddings, and collection name

# Initialize the Qdrant vector store from langchain

db = Qdrant(

client=client,

embeddings=df['encoded\_news'].tolist(), # Use the generated embeddings

collection\_name=collection\_name

)

logging.info(f"Qdrant vector store initialized: {db}") # Prints the database object information

2024-04-26 16:27:55 - INFO - QdrantClient initialized: <qdrant\_client.qdrant\_client.QdrantClient object at 0x0000025DABB59D60>

2024-04-26 16:27:55 - INFO - #################################

D:\Desktop\Superteams AI\Task 2 Vector Databases can help with Anomaly Detection\Vector-Databases-can-help-with-Anomaly-Detection\venv\Lib\site-packages\langchain\_community\vectorstores\qdrant.py:150: UserWarning: `embeddings` should be an instance of `Embeddings`.Using `embeddings` as `embedding\_function` which is deprecated

warnings.warn(

2024-04-26 16:27:55 - INFO - Qdrant vector store initialized: <langchain\_community.vectorstores.qdrant.Qdrant object at 0x0000025DAD359F70>

def similarity\_search\_with\_score(query, k=2):

query\_embedding = generate\_embedding(query)[0].tolist()

search\_results = qdrant\_client.search(

collection\_name=collection\_name,

query\_vector=query\_embedding,

limit=k,

with\_payload=True,

with\_vectors=False

)

return search\_results

query = "Stock market crashes due to unexpected event"

search\_results = similarity\_search\_with\_score(query=query, k=5)

for result in search\_results:

doc\_id = result.id

score = result.score

payload = result.payload # The payload should contain your text or a reference to it.

# Assuming the payload contains a field 'input' where the text is stored

doc\_content = payload.get('output', 'No content available')

# Print the similarity score and document content

logging.info({"score": score, "doc\_id": doc\_id, "content": doc\_content})

2024-04-26 16:33:48 - INFO - HTTP Request: POST http://localhost:6333/collections/Finance%20Outliers/points/search "HTTP/1.1 200 OK"

2024-04-26 16:33:48 - INFO - {'score': 0.296198, 'doc\_id': 0, 'content': "Kotak Mahindra Bank's loan, deposit growth may be impacted after RBI curbs Kotak Mahindra Bank's loan, deposit growth may be impacted after RBI curbs\n Kotak Mahindra Bank’s loan and deposit growth are likely to be affected after the Reserve Bank of India (RBI) asked the private-sector lender not to take on board new customers through the bank’s online and mobile banking channels and not to issue any new credit cards, according to analysts. Photograph: Adnan Abidi/Reuters The bank’s share price fell 10.85 per cent on Thursday to close the day at Rs 1,643 on the BSE. The RBI’s action came after market hours on ...\nRediff.com, 1 hour(s) ago\nAlso from:"}

2024-04-26 16:33:48 - INFO - {'score': 0.28810832, 'doc\_id': 9, 'content': 'Sensex revisits 74K; Nifty climbs 168 points Sensex revisits 74K; Nifty climbs 168 points\n Rising for the fifth straight session, equity benchmark Sensex rallied nearly 500 points to reclaim the 74,000 mark while the Nifty closed above the 22,550 level on Thursday, driven by heavy buying in banking, financial and metal stocks. Photograph: Shailesh Andrade/Reuters Recovering after a sell-off in early trade, the 30-share BSE Sensex climbed 486.50 points or 0.66 per cent to settle at 74,339.44. During the day, it surged 718.31 points or 0.97 per cent to 74,571.25. The NSE Nifty ...\nRediff.com, 23 hour(s) ago\nAlso from: Rediff.com'}

2024-04-26 16:33:48 - INFO - {'score': 0.26815844, 'doc\_id': 6, 'content': 'Good Q4, commentary perk up ICICI Lombard General Insurance stock Good Q4, commentary perk up ICICI Lombard General Insurance stock\n Investment yields could be around 8.1 per cent in FY25 rising to 8.5 per cent in FY26 Photograph: Courtesy, ICICI Lombard ICICI Lombard General Insurance Company reported financial improvement and optimistic commentary in Q4FY24. It reported 17 per cent year-on-year (YoY) growth in Gross Written Premium (GWP) and 115 bps improvement in the Combined Ratio (COR) in FY24, and improved COR guidance with COR going from 104.5 per cent in FY23 to 103.3 per cent in FY24, 102.4 per cent in FY25 and ...\nRediff.com, 5 hour(s) ago\nAlso from:'}

2024-04-26 16:33:48 - INFO - {'score': 0.23023885, 'doc\_id': 4, 'content': "Tech Mahindra jumps over 12% in opening trade Tech Mahindra jumps over 12% in opening trade\n Equity benchmark indices climbed in early trade on Friday, extending their rally for the sixth day running, on heavy buying in Tech Mahindra and firm trends in Asian markets. Photograph: Adnan Abidi/Reuters The 30-share BSE Sensex climbed 176.47 points to 74,515.91 in early trade. The NSE Nifty went up by 50.05 points to 22,620.40. \xa0 From the Sensex basket, Tech Mahindra jumped over 12.50 per cent after the IT services company's CEO outlined an ambitious three-year roadmap to accelerate ...\nRediff.com, 5 hour(s) ago\nAlso from:"}

2024-04-26 16:33:48 - INFO - {'score': 0.21434154, 'doc\_id': 1, 'content': 'Above-normal monsoon likely to ease food prices: FinMin Above-normal monsoon likely to ease food prices: FinMin\n With the prediction of an above normal monsoon in 2024, the government is expecting food prices to come down, the finance ministry’s monthly economic report for March has said. Photograph: Amit Dave/Reuters The report, released on Thursday, said robust foreign inflows and comfortable trade deficits were expected to keep the rupee within a comfortable range. “Further easing of food prices is on the anvil as IMD (India Meteorological Department) has predicted above-normal rainfall ...\nRediff.com, 3 hour(s) ago\nAlso from:'}

### **Modify This Function to Give Contacted String**

def similarity\_search\_with\_score(query, k=2):

query\_embedding = generate\_embedding(query)[0].tolist()

search\_results = qdrant\_client.search(

collection\_name=collection\_name,

query\_vector=query\_embedding,

limit=k,

with\_payload=True,

with\_vectors=False

)

# Extract the document content from the payload and include it in the results

results\_with\_content = []

for result in search\_results:

doc\_id = result.id

score = result.score

payload = result.payload # The payload should contain your text or a reference to it.

# Extract the document content from the payload

doc\_content = payload.get('output', 'No content available')

results\_with\_content.append((score, doc\_content))

# Sort the results based on the similarity score in descending order

sorted\_results = sorted(results\_with\_content, key=lambda x: x[0], reverse=True)

# Concatenate the content of the top k results

concatenated\_content = ' '.join([content for \_, content in sorted\_results[:k]])

return concatenated\_content

query = "Stock market crashes due to unexpected event"

outlier\_paragraph = similarity\_search\_with\_score(query=query, k=5)

# Print the concatenated content

logging.info({"concatenated\_content": outlier\_paragraph})

**2024-04-26 16:36:50 - INFO - HTTP Request: POST http://localhost:6333/collections/Finance%20Outliers/points/search "HTTP/1.1 200 OK"**

**2024-04-26 16:36:50 - INFO - {'concatenated\_content': "Kotak Mahindra Bank's loan, deposit growth may be impacted after RBI curbs Kotak Mahindra Bank's loan, deposit growth may be impacted after RBI curbs\n Kotak Mahindra Bank’s loan and deposit growth are likely to be affected after the Reserve Bank of India (RBI) asked the private-sector lender not to take on board new customers through the bank’s online and mobile banking channels and not to issue any new credit cards, according to analysts. Photograph: Adnan Abidi/Reuters The bank’s share price fell 10.85 per cent on Thursday to close the day at Rs 1,643 on the BSE. The RBI’s action came after market hours on ...\nRediff.com, 1 hour(s) ago\nAlso from: Sensex revisits 74K; Nifty climbs 168 points Sensex revisits 74K; Nifty climbs 168 points\n Rising for the fifth straight session, equity benchmark Sensex rallied nearly 500 points to reclaim the 74,000 mark while the Nifty closed above the 22,550 level on Thursday, driven by heavy buying in banking, financial and metal stocks. Photograph: Shailesh Andrade/Reuters Recovering after a sell-off in early trade, the 30-share BSE Sensex climbed 486.50 points or 0.66 per cent to settle at 74,339.44. During the day, it surged 718.31 points or 0.97 per cent to 74,571.25. The NSE Nifty ...\nRediff.com, 23 hour(s) ago\nAlso from: Rediff.com Good Q4, commentary perk up ICICI Lombard General Insurance stock Good Q4, commentary perk up ICICI Lombard General Insurance stock\n Investment yields could be around 8.1 per cent in FY25 rising to 8.5 per cent in FY26 Photograph: Courtesy, ICICI Lombard ICICI Lombard General Insurance Company reported financial improvement and optimistic commentary in Q4FY24. It reported 17 per cent year-on-year (YoY) growth in Gross Written Premium (GWP) and 115 bps improvement in the Combined Ratio (COR) in FY24, and improved COR guidance with COR going from 104.5 per cent in FY23 to 103.3 per cent in FY24, 102.4 per cent in FY25 and ...\nRediff.com, 5 hour(s) ago\nAlso from: Tech Mahindra jumps over 12% in opening trade Tech Mahindra jumps over 12% in opening trade\n Equity benchmark indices climbed in early trade on Friday, extending their rally for the sixth day running, on heavy buying in Tech Mahindra and firm trends in Asian markets. Photograph: Adnan Abidi/Reuters The 30-share BSE Sensex climbed 176.47 points to 74,515.91 in early trade. The NSE Nifty went up by 50.05 points to 22,620.40. \xa0 From the Sensex basket, Tech Mahindra jumped over 12.50 per cent after the IT services company's CEO outlined an ambitious three-year roadmap to accelerate ...\nRediff.com, 5 hour(s) ago\nAlso from: Above-normal monsoon likely to ease food prices: FinMin Above-normal monsoon likely to ease food prices: FinMin\n With the prediction of an above normal monsoon in 2024, the government is expecting food prices to come down, the finance ministry’s monthly economic report for March has said. Photograph: Amit Dave/Reuters The report, released on Thursday, said robust foreign inflows and comfortable trade deficits were expected to keep the rupee within a comfortable range. “Further easing of food prices is on the anvil as IMD (India Meteorological Department) has predicted above-normal rainfall ...\nRediff.com, 3 hour(s) ago\nAlso from:"}**

## **RAG Using Llama 3**

Using the concatenated result, generate a paragraph on anomalies that allows you to read everything at once. Article [5] explains how to set up Llama 2 on your local system and run it. The code in [2] that you just git cloned includes the PS1 script to run it. The code in [2] contains a file called llama3backend, which makes llama3 a simple function call away. This file automatically calls llama2 GGUF, which is a quantized version of llama3.

query = "Stock market crashes due to unexpected event ? Summarize this in one paragraph "

from llama3backend import generate\_text

RAG\_answer = generate\_text(str(query + outlier\_paragraph)[:512])

2024-04-26 16:45:20 - INFO - {'RAG\_answer': " debit cards or other payment instruments.\nThe RBI has taken this step as a precautionary measure to maintain financial stability in the country. The RBI has also asked Kotak Mahindra Bank to review its lending policies and ensure that they are in line with the RBI’s guidelines.\nIn conclusion, Kotak Mahindra Bank's loan and deposit growth may be impacted after the Reserve Bank of India (RBI) asked the private-sector lender not to take on board new customers through the bank’s online and mobile banking channels and not to issue any new credit cards, debit cards or other payment instruments. The RBI has taken this step as a precautionary measure to maintain financial stability in the country. The RBI has also asked Kotak Mahindra Bank to review its lending policies and ensure that they are in line with the RBI’s guidelines.\nIn conclusion, Kotak Mahindra Bank's loan and deposit growth may be impacted after the Reserve Bank of India (RBI) asked the private-sector lender not to take on board new customers through the bank’s online and mobile banking channels and not to issue any new credit cards, debit cards or other payment instruments. The RBI has taken this step as a precautionary measure to maintain financial stability in the country. The RBI has also asked Kot"}

## **Anomaly Detection**

This method is used for larger datasets, so we will be using [6]. We will be loading it

https://www.kaggle.com/datasets/rmisra/news-category-dataset

import pandas as pd

# Load the CSV file into a pandas DataFrame

df = pd.read\_json("Dataset/News\_Category\_Dataset\_v3.json" , lines=True)

df.head()

Take the first 1000 rows.

# Select the first 1000 rows of the DataFrame

df = df.head(1000)

Create embeddings like before.

# Generate embeddings for the 'input' column

df['encoded\_news'] = df['news'].apply(lambda x: generate\_embedding(x)[0].tolist())

df.head()

### **Apply t-SNE for Dimensionality Reduction**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.manifold import TSNE

X = np.array(df['encoded\_news'].tolist())

# Increase the perplexity value (default is 30)

tsne = TSNE(random\_state=0, n\_iter=1000, perplexity=500)

tsne\_results = tsne.fit\_transform(X)

df\_tsne = pd.DataFrame(tsne\_results, columns=['TSNE1', 'TSNE2'])

df\_tsne['Class Name'] = df['category'] # Using 'title' as a placeholder for 'Class Name'

df\_tsne['news'] = df['news']

# df\_tsne['encoded\_news'] = df['encoded\_news']

df\_tsne.head()

# Plot t-SNE results

fig, ax = plt.subplots(figsize=(8, 6))

sns.set\_style('darkgrid', {"grid.color": ".6", "grid.linestyle": ":"})

sns.scatterplot(data=df\_tsne, x='TSNE1', y='TSNE2', hue='Class Name', palette='Set2')

sns.move\_legend(ax, "upper left", bbox\_to\_anchor=(1, 1))

plt.title('Scatter plot of news using t-SNE')

plt.xlabel('TSNE1')

plt.ylabel('TSNE2')

plt.show()

### **Outlier Detection**

# Function to get centroids of each class

def get\_centroids(df\_tsne):

centroids = df\_tsne.drop(columns=['news']).groupby('Class Name').mean()

return centroids

centroids = get\_centroids(df\_tsne)

# Function to detect outliers

def calculate\_euclidean\_distance(p1, p2):

return np.sqrt(np.sum(np.square(p1 - p2)))

def detect\_outlier(df, emb\_centroids, radius):

outlier\_indices = []

for idx, row in df.iterrows():

class\_name = row['Class Name']

dist = calculate\_euclidean\_distance(np.array([row['TSNE1'], row['TSNE2']]),

np.array([emb\_centroids.loc[class\_name, 'TSNE1'],

emb\_centroids.loc[class\_name, 'TSNE2']]))

if dist > radius:

outlier\_indices.append(idx)

return outlier\_indices

# Assuming df\_tsne and centroids are already defined DataFrames

range\_ = np.arange(0.01, 1.0, 0.05).round(decimals=2).tolist()

outliers\_list = []

for i in range\_:

outliers = detect\_outlier(df\_tsne, centroids, i)

outliers\_list.append(outliers)

# Combine all outlier indices into a single list

all\_outliers = [idx for sublist in outliers\_list for idx in sublist]

# Update the 'Outlier' column in df\_tsne

df\_tsne['Outlier'] = df\_tsne.index.isin(all\_outliers)

df\_tsne.head()

# Assuming df\_tsne and centroids are already defined DataFrames

range\_ = np.arange(0.01, 1.0, 0.05).round(decimals=2).tolist()

outliers\_list = []

for i in range\_:

outliers = detect\_outlier(df\_tsne, centroids, i)

outliers\_list.append(outliers)

# Combine all outlier indices into a single list

all\_outliers = [idx for sublist in outliers\_list for idx in sublist]

# Update the 'Outlier' column in df\_tsne

df\_tsne['Outlier'] = df\_tsne.index.isin(all\_outliers)

df\_tsne.head()

# Filter out rows where 'Outlier' is False

df\_tsne = df\_tsne[df\_tsne['Outlier'] == True]

num\_outliers = [len(outliers) for outliers in outliers\_list]

import matplotlib.pyplot as plt

# Plot range\_ and num\_outliers

fig = plt.figure(figsize=(14, 8))

plt.rcParams.update({'font.size': 12})

plt.bar(list(map(str, range\_)), num\_outliers)

plt.title("Number of outliers vs. distance of points from centroid")

plt.xlabel("Distance")

plt.ylabel("Number of outliers")

for i in range(len(range\_)):

plt.text(i, num\_outliers[i], num\_outliers[i], ha='center')

plt.show()

def get\_outlier\_texts(df, class\_name):

# Filter the DataFrame to get outliers of the specified category

outliers = df[(df['Class Name'] == class\_name) & df['Outlier']]

# Extract the outlier texts

outlier\_texts = outliers['news'].tolist()

return outlier\_texts

# Example usage:

outlier\_texts = get\_outlier\_texts(df\_tsne, 'TECH')

for idx, text in enumerate(outlier\_texts, start=1):

print(f"Outlier {idx}: {text}\n")

Outlier 1: Twitch Bans Gambling Sites After Streamer Scams Folks Out Of $200,000 One man's claims that he scammed people on the platform caused several popular streamers to consider a Twitch boycott.

Outlier 2: TikTok Search Results Riddled With Misinformation: Report A U.S. firm that monitors false online claims reports that searches for information about prominent news topics on TikTok are likely to turn up results riddled with misinformation.

Outlier 3: Citing Imminent Danger Cloudflare Drops Hate Site Kiwi Farms Cloudflare CEO Matthew Prince had previously resisted calls to block the site.

Outlier 4: Instagram And Facebook Remove Posts Offering Abortion Pills Facebook and Instagram began removing some of these posts, just as millions across the U.S. were searching for clarity around abortion access.

Outlier 5: Google Engineer On Leave After He Claims AI Program Has Gone Sentient Artificially intelligent chatbot generator LaMDA wants “to be acknowledged as an employee of Google rather than as property," says engineer Blake Lemoine.

Outlier 6: Facebook Is Still Allowing Mug Shots Even Though They Can Ruin Lives When an individual’s mug shot goes viral on Facebook, they are often subjected to extreme harassment and struggle to find stable housing and employment.

Outlier 7: Ex-Twitter CEO Dings Elon Musk For Attacks On Twitter's Top Lawyer A one-sided feud between Musk and Vijaya Gadde has turned even uglier.

Outlier 8: Investor Sues Elon Musk Over His Delayed Twitter Filing Marc Rasella says he sold shares of Twitter at “artificially deflated prices,” unaware that Musk had made a large purchase in the social media platform.

def get\_outlier\_paragraph(df, class\_name):

# Filter the DataFrame to get outliers of the specified category

outliers = df[(df['Class Name'] == class\_name) & df['Outlier']]

# Extract the outlier texts

outlier\_texts = outliers['news'].tolist()

# Concatenate all outlier texts into one paragraph

outlier\_paragraph = ' '.join(outlier\_texts)

return outlier\_paragraph

# Example usage:

outlier\_paragraph = get\_outlier\_paragraph(df\_tsne, 'TECH')

print(outlier\_paragraph)

Twitch Bans Gambling Sites After Streamer Scams Folks Out Of $200,000 One man's claims that he scammed people on the platform caused several popular streamers to consider a Twitch boycott. TikTok Search Results Riddled With Misinformation: Report A U.S. firm that monitors false online claims reports that searches for information about prominent news topics on TikTok are likely to turn up results riddled with misinformation. Citing Imminent Danger Cloudflare Drops Hate Site Kiwi Farms Cloudflare CEO Matthew Prince had previously resisted calls to block the site. Instagram And Facebook Remove Posts Offering Abortion Pills Facebook and Instagram began removing some of these posts, just as millions across the U.S. were searching for clarity around abortion access. Google Engineer On Leave After He Claims AI Program Has Gone Sentient Artificially intelligent chatbot generator LaMDA wants “to be acknowledged as an employee of Google rather than as property," says engineer Blake Lemoine. Facebook Is Still Allowing Mug Shots Even Though They Can Ruin Lives When an individual’s mug shot goes viral on Facebook, they are often subjected to extreme harassment and struggle to find stable housing and employment. Ex-Twitter CEO Dings Elon Musk For Attacks On Twitter's Top Lawyer A one-sided feud between Musk and Vijaya Gadde has turned even uglier. Investor Sues Elon Musk Over His Delayed Twitter Filing Marc Rasella says he sold shares of Twitter at “artificially deflated prices,” unaware that Musk had made a large purchase in the social media platform.

### **RAG Using Llama 3**

query = "write a summary for this ? "

print(str(query + outlier\_paragraph)[:512])

write a summary for this ? Twitch Bans Gambling Sites After Streamer Scams Folks Out Of $200,000 One man's claims that he scammed people on the platform caused several popular streamers to consider a Twitch boycott. TikTok Search Results Riddled With Misinformation: Report A U.S. firm that monitors false online claims reports that searches for information about prominent news topics on TikTok are likely to turn up results riddled with misinformation. Citing Imminent Danger Cloudflare Drops Hate Site Kiwi Fa

from llama3backend import generate\_text

RAG\_answer = generate\_text(str(query + outlier\_paragraph)[:512])

RAG\_answer

'... (read more) ...r, Which Was Linked To The Christchurch Mosque Massacre. Cloudflare, a content delivery network (CDN), has dropped its support for hate site Kiwi Farms, which was linked to the Christchurch mosque massacre. The decision comes after Cloudflare faced intense pressure from human rights groups and other organizations to sever ties with the hate site. In a statement, Cloudflare said that it had "re-evaluated" its relationship with Kiwi Farms and had decided to terminate its support for the site. The company said that it would continue to provide services to other websites and organizations that promote hate speech or other forms of discrimination. (read more) ...r, Which Was Linked To The Christchurch Mosque Massacre. Cloudflare, a content delivery network (CDN), has dropped its support for hate site Kiwi Farms, which was linked to the Christchurch mosque massacre. The decision comes after Cloudflare faced intense pressure from human rights groups and other organizations to sever ties with the hate site. In a statement, Cloudflare said that it had "re-evaluated" its relationship with Kiwi Farms and had decided to terminate its support for the site. The company said that it would continue to provide services to other websites and organizations that'

## **References**

[1] “Realtime Market News. Live BSE, NSE, Stock Prices, Expert Stock Advice, Share Market Updates : Rediff.com.” Accessed: Apr. 18, 2024. [Online]. Available: https://money.rediff.com/news

[2] A. Pranav, “How to Install and Run Llama2 Locally on Windows for Free,” Medium. Accessed: Apr. 18, 2024. [Online]. Available: https://medium.com/@AyushmanPranav/how-to-install-and-run-llama2-locally-on-windows-for-free-05bd5032c6e3

[3] “MaziyarPanahi/Meta-Llama-3-8B-Instruct-GGUF at main.” Accessed: Apr. 26, 2024. [Online]. Available: https://huggingface.co/MaziyarPanahi/Meta-Llama-3-8B-Instruct-GGUF/tree/main

[4] A. Pranav, “Steps to Monitoring DSPy-Qdrant Powered RAG with Prometheus or Grafana,” DevOps.dev. Accessed: Apr. 18, 2024. [Online]. Available: https://blog.devops.dev/steps-to-monitoring-dspy-qdrant-powered-rag-with-prometheus-or-grafana-b642335cbd50

[5] “heathbrew/Vector-Databases-can-help-with-Anomaly-Detection: Pinecone , reddiffmoney financial dataset.” Accessed: Apr. 18, 2024. [Online]. Available: https://github.com/heathbrew/Vector-Databases-can-help-with-Anomaly-Detection

[6] “News Category Dataset.” Accessed: Apr. 18, 2024. [Online]. Available: https://www.kaggle.com/datasets/rmisra/news-category-dataset