Data Collection, Classification, and Transformation

Instructors:

Lingzi Hong & Xiaoying Song

Presenter Profile

- Presenter: Lingzi Hong
- Assistant Professor in Data Science at the University of North Texas
- Research Interest:
  - Computational Linguistics
  - Human-Centered Computing
  - Data Literacy
  - Social Media Analysis

Presenter Profile

- Presenter: Xiaoying Song
- PhD student in Information Science at the University of North Texas
- Research Interest:
  - Large Language Models
  - Online Misbehavior
  - Counter Speech
  - User Information Behavior

Outline

- Theoretical Part (30 minutes)
  - APIs
  - Classifications with Large Language Models (LLMs)
  - Data Transformation
- Demonstrations (20 minutes)
  - Data Collection from Semantic Scholar
  - Classification with Llama2
  - Data Analysis with Python
- Q&A (10 minutes)

THEORETICAL PART

Motivation

- Service Enhancement
  - Librarians can better understand user needs and preferences by analyzing posts and comments on social media and other digital platforms.
  - This information can be used to develop strategies to improve user engagement, develop targeted programs based on user needs.

- Resource Management
  - Librarians can understand how data is managed and how scholars communicate by analyzing data from research platforms, such as the Semantic Scholar.
  - These insights can enhance librarians' ability to support academic research more effectively.

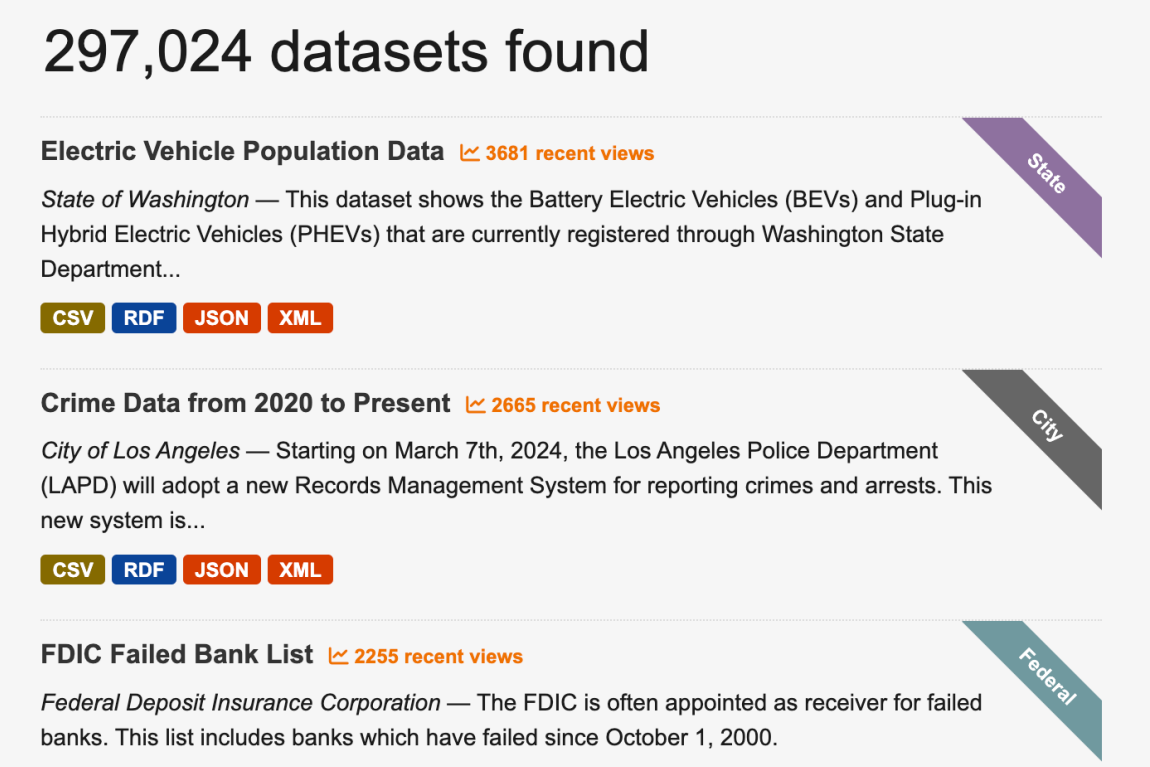
Motivation

- Provide Data Literacy Service
  - Data literacy is important for library patrons, such as college students and researchers, in their informed decision-making and research practices.
  - Librarians play a key role in educating patrons about data literacy.
  - Knowledge of how data is gathered and used can help librarians teach:
    - how to identify trustworthy data online
    - how to collect data
    - data privacy issues

Methods to Collect Online Data

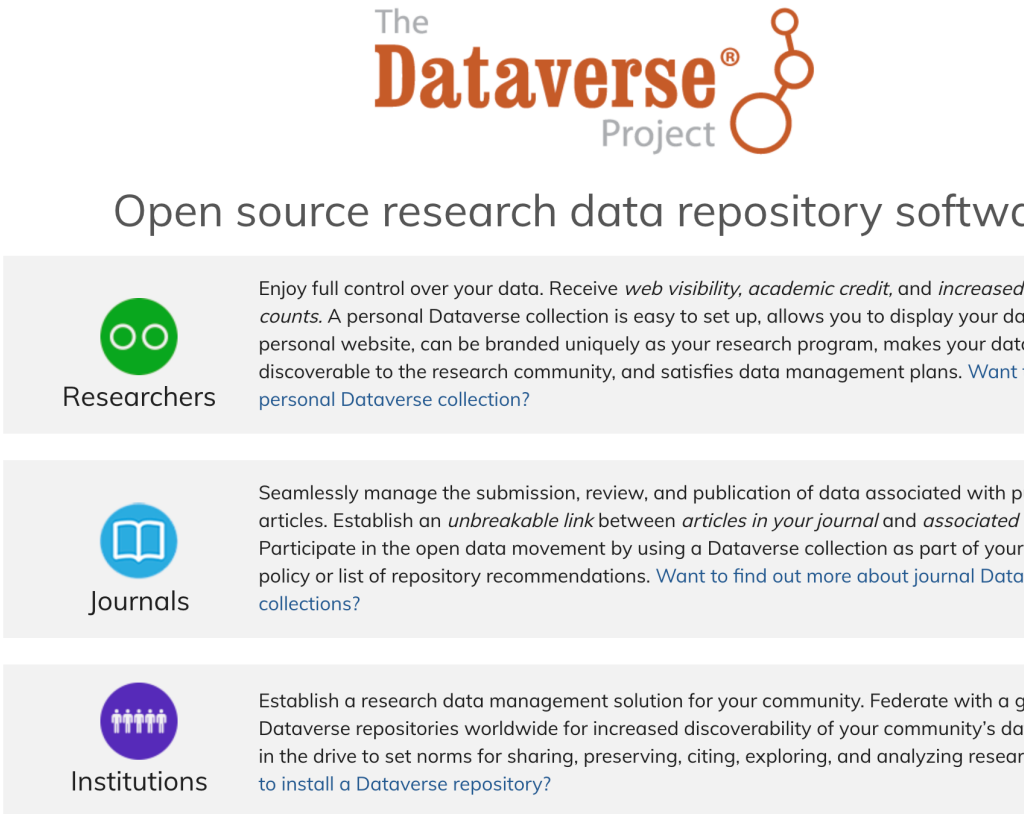
- Export data files from reliable sources
  - Open government data portals: Data.gov
  - Open-source research data repository: Dataverse
- Webpage Scrapping
  - Programmatically collect data from web pages
- Website Application Programming Interfaces (APIs)
  - Collect metadata of resources
  - Collect data on public posts and user interactions

Data Portals



- **Data.gov** aims to improve public access to high value, machine-readable datasets generated by the Executive Branch of the Federal Government.
- The site is a repository for Federal, state, local, and tribal government information made available to the public.

Data Repository



- **The Dataverse** is an open-source web application to share, preserve, cite, explore and analyze research data. Researchers, data authors, publishers, data distributors, and affiliated institutions all receive appropriate credit via a data citation with a persistent identifier.

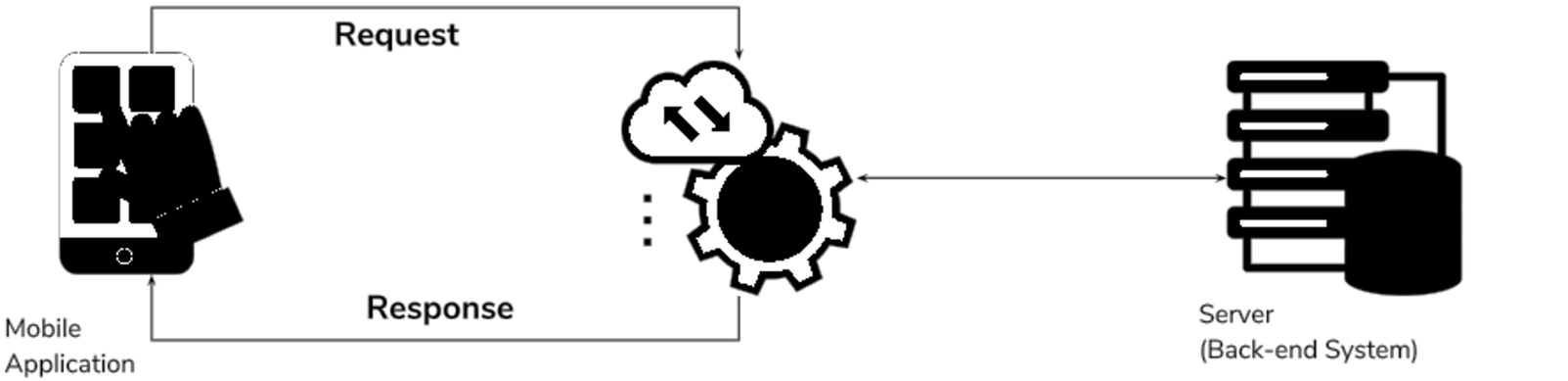
Web Page Scrapping

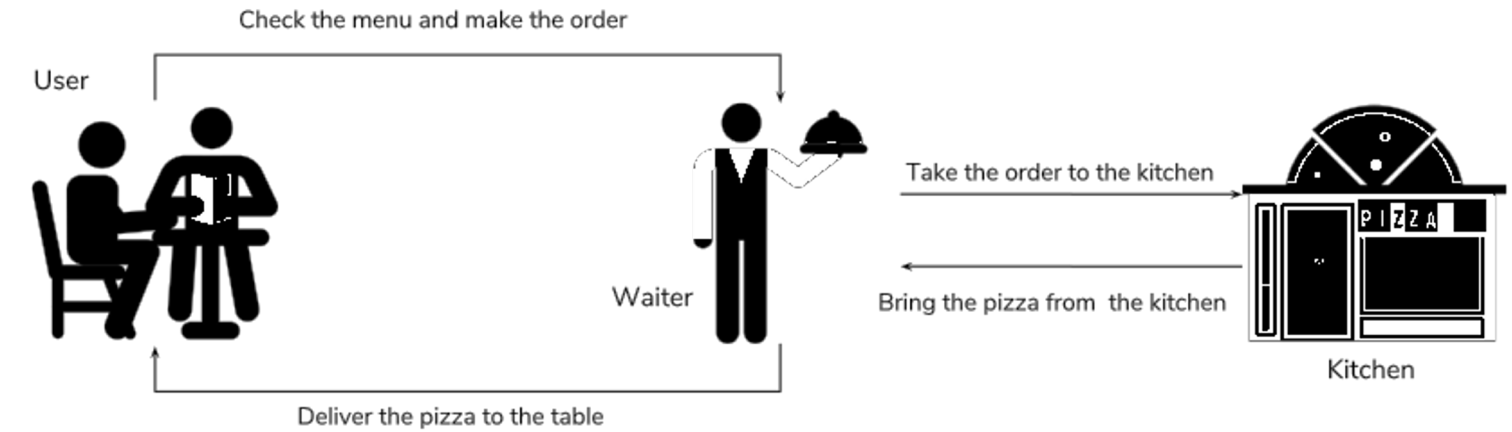
- Web scraping involves programmatically gathering data from web pages using tools like BeautifulSoup or Selenium in Python.
- Pros:
  - Flexibility: Can retrieve any data that can be viewed in a browser.
  - Control: Provides granular control over what data is extracted and how it's gathered.
- Cons:
  - Fragility: Web scraping is susceptible to breaking if the structure of the web page changes.
  - Legal and Ethical Issues: Scraping data from websites without permission may violate terms of service or legal regulations.

APIs

- An API is a set of rules and protocols for building and interacting with software applications. A web API allows programs to request data from services and respond with data in a structured format.
- **How It Works:** APIs act as gateways for accessing the functionalities or data of an application, server, or other services. They provide data in a structured format, such as JSON or XML, following specific requests made over the web (using protocols like HTTP).
- **Use Cases:** APIs are typically used to interact with web services or to integrate different software components. For example, social media platforms and weather services often provide APIs to allow developers to access their services programmatically.

How it works





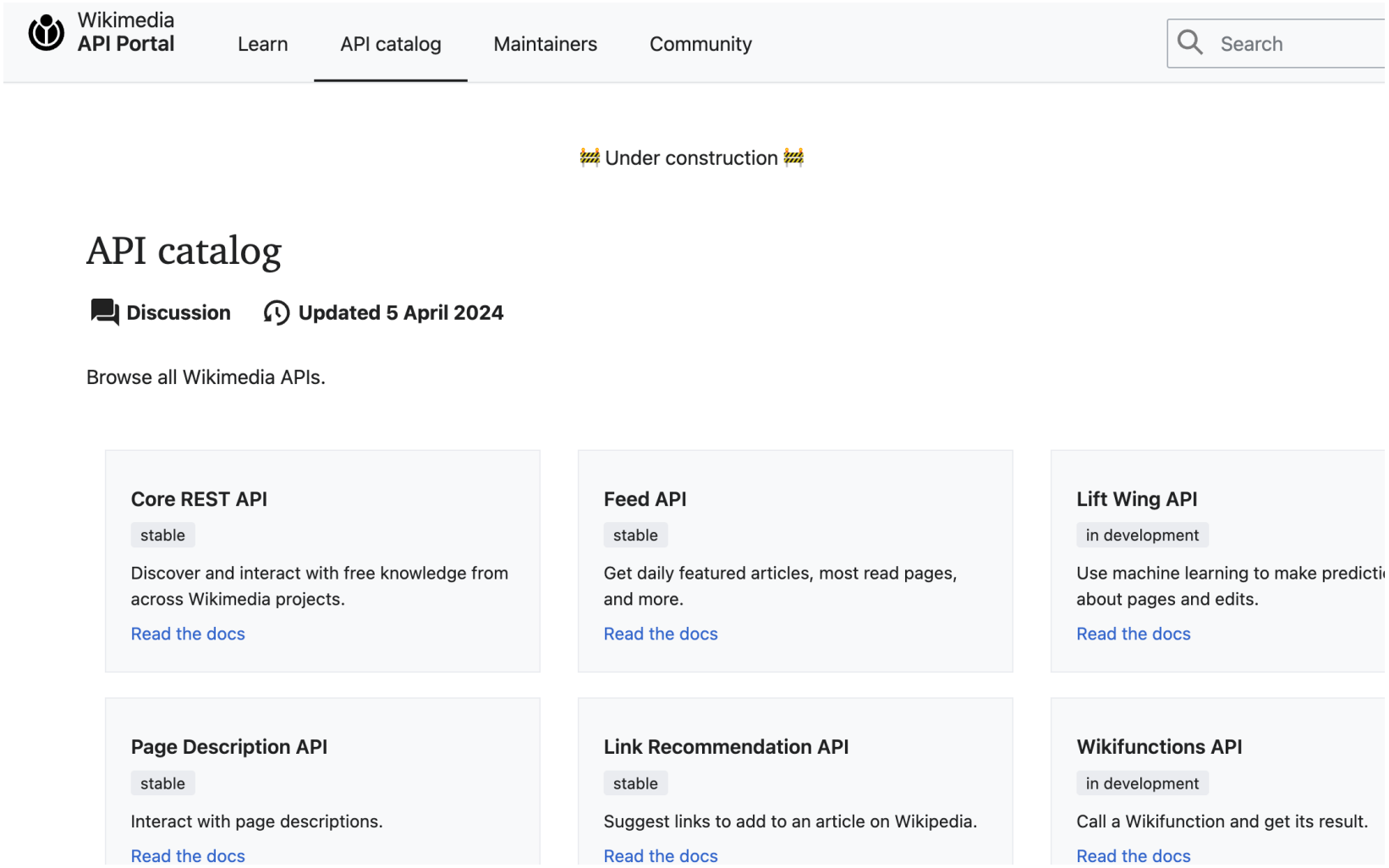
Source: https://medium.com/@ama.thanu/what-is-an-api-how-does-it-work-f4ea552d741f

Why should we care?

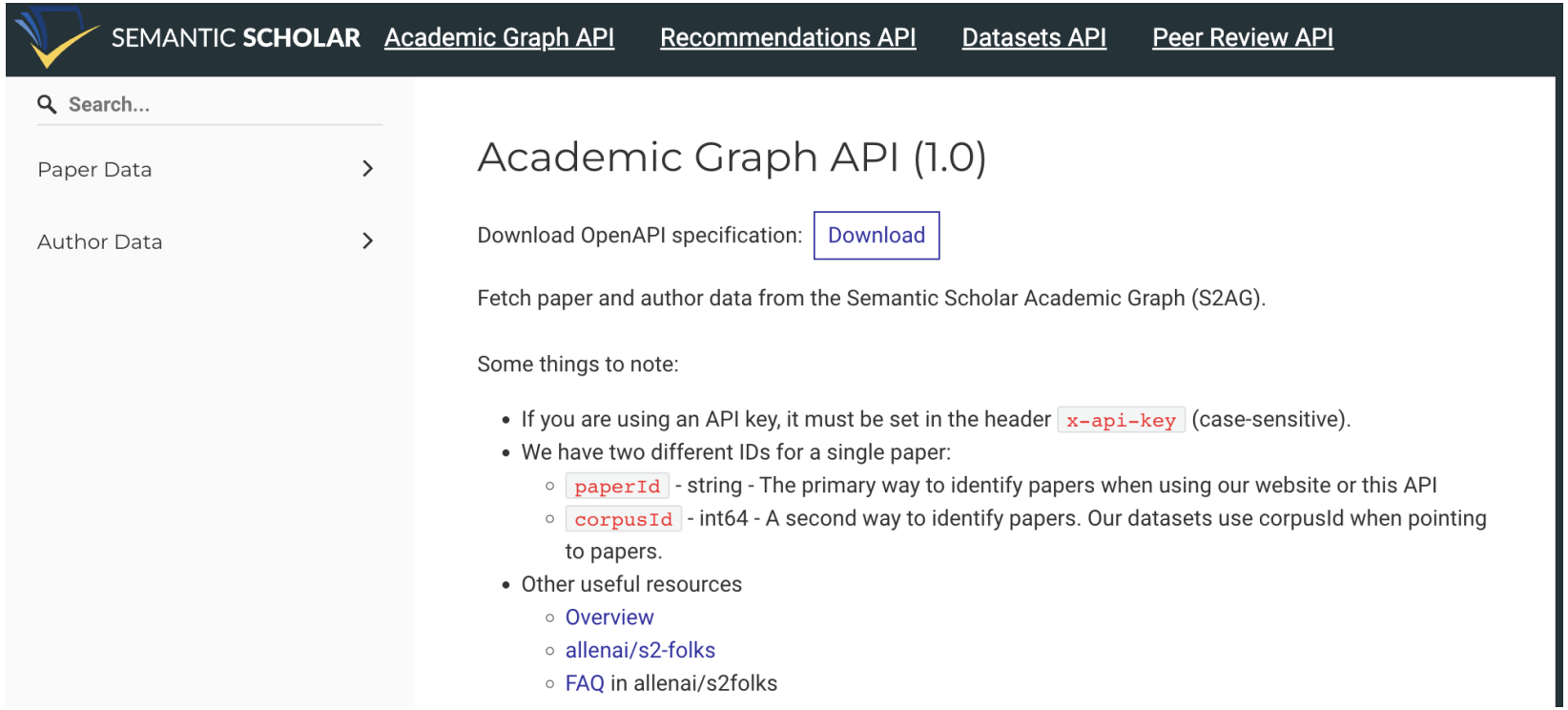
- A lot of companies allow users to freely access data from their websites by means of APIs. It is important to have some exposure on how to work and get data with APIs.

Source: https://medium.com/nerd-for-tech/api-integration-the-top-api-benefits-for-your-business-1aa31225efbf

Examples: Wikimedia APIs

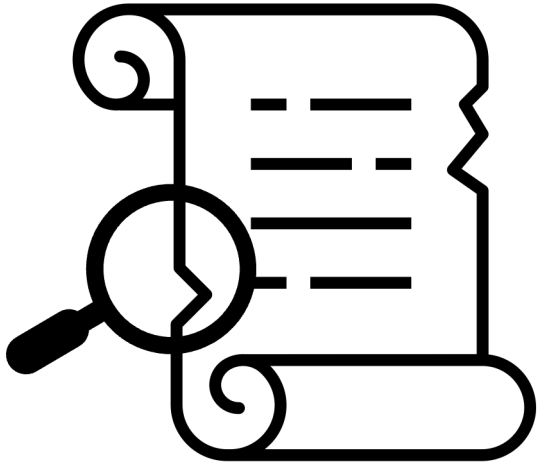


Examples: Semantic Scholar APIs



Why use Semantic Scholar API?

- The Semantic Scholar REST API allows you to find and explore scientific publication data about authors, papers, citations, venues, and more. The API is organized into the following services:
  - Academic Graph
  - Recommendations
  - Datasets
  - Conference Peer Review



https://www.semanticscholar.org/product/api

Working with Web APIs

- We use web APIs to access web content and data.

- How to work with web APIs?
  - Read the API's documentation
  - Read terms of service (what you are allowed or not allowed to do)
  - You may need authentication for access to data

Source: https://magepow.com/blog/what-is-web-api-how-does-web-api-work/

Natural Language Processing

Natural language processing (NLP) is the ability of a computer program to understand human language as it's spoken and written -- referred to as natural language.

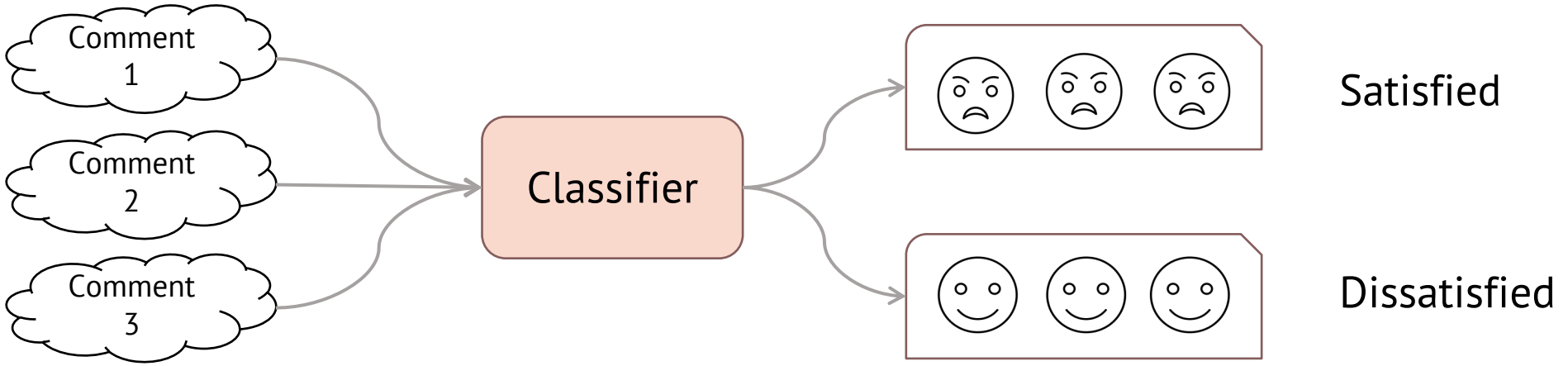
Applications of NLP in text classification

- Sentiment Analysis
- Spam Detection
- Language Identification
- News Categorization

What is Text Classification?

Text classification in NLP involves categorizing and assigning predefined labels or categories to text documents, sentences, or phrases based on their content. Text classification aims to automatically determine the class or category to which a piece of text belongs (Analytics Vidhya, 2020).

Text classification algorithms analyze the features and patterns within the text to make accurate predictions about its category, enabling machines to organize, filter, and understand large volumes of textual data.



Analytics Vidhya. (2020, December). Understanding text classification in NLP with a movie review example. Retrieved from https://www.analyticsvidhya.com/blog/2020/12/understanding-text-classification-in-nlp-with-movie-review-example-example/

Zero-Shot Classification

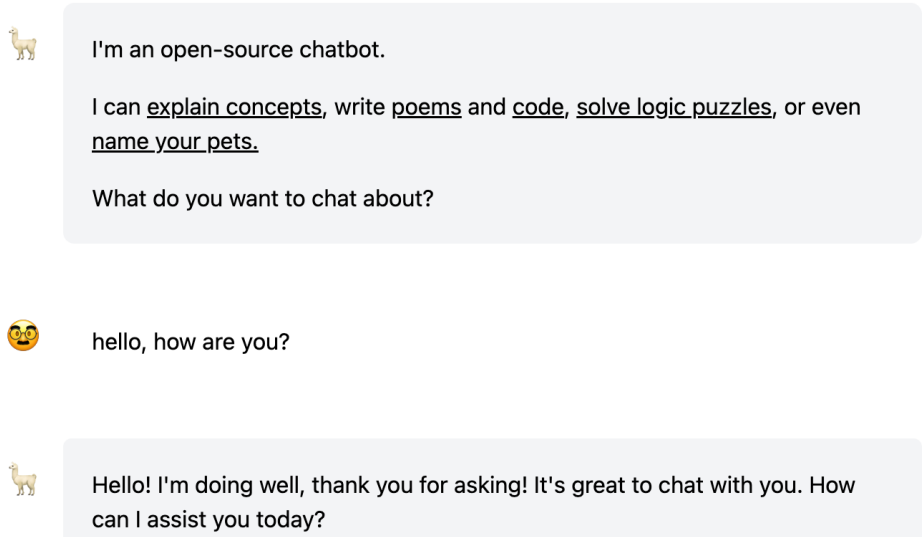
- Zero Shot Classification is the task of predicting a class that wasn't seen by the model during training.
- In zero-shot classification, we provide the model with a prompt and a sequence

of text that describes what we want our model to do.

Source: https://huggingface.co/tasks/zero-shot-classification

Llama 2 for Classification

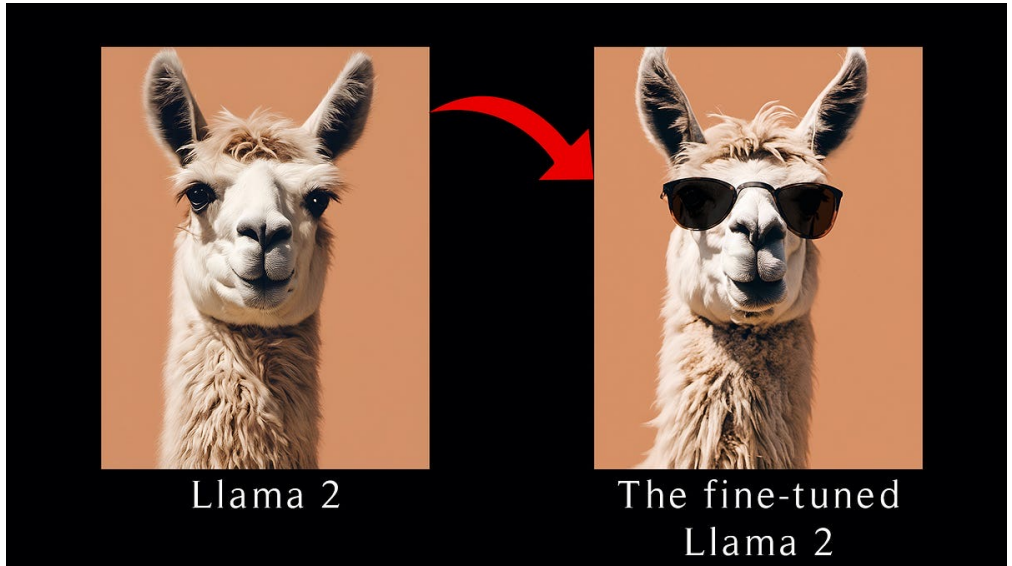
- LLaMA2, introduced by Meta in 2023, is an open source large language model (LLMs). It is a part of the LLaMA (Language Large Model) family, which encompasses a range of models with varying capacities, from 7 billion to 70 billion parameters.
- LLaMA2 also features models specifically fine-tuned for certain applications. For example, LLaMA Chat, optimized for dialogue use cases, has been trained on over 1 million human annotations to enhance its conversational abilities.



Source:
https://llama-2.ai/llama-2-model-details/
https://www.run.ai/guides/generative-ai/llama-2-fine-tuning

LLM Fine-tuning

- Fine-tuning large language models involves adapting the pre-trained model to perform specific tasks or understand particular domains better. This is achieved by training the model on a new dataset that is more focused on the desired task or domain.
- Supervised Fine-Tuning (SFT) is a process where a pre-trained language model is further trained (fine-tuned) on a smaller, task-specific dataset under human supervision. For instance, if LLaMA2 needs to be specialized for medical data analysis, it would undergo SFT on a dataset comprising medical texts, patient records, and related literature.



Source:
https://ramseyelbasheer.io/2023/08/31/a-beginners-guide-to-llm-fine-tuning/

https://www.run.ai/guides/generative-ai/llama-2-fine-tuning

DEMONSTRATIONS

Get data with APIs
Semantic Scholar API

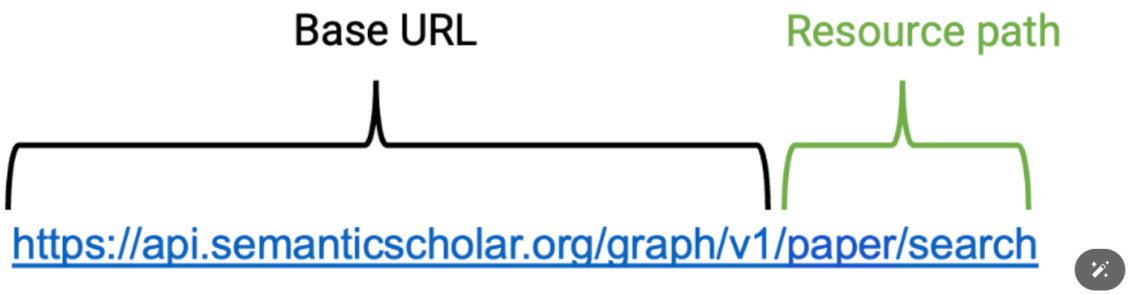
- Consult the API documentation
  - Every API should provide you with some sort of documentation to start. Usually, there will be a reference section that provides the various objects, parameters, and endpoints you can access.
  - https://www.semanticscholar.org/product/api/tutorial

How do I know the endpoint URL?

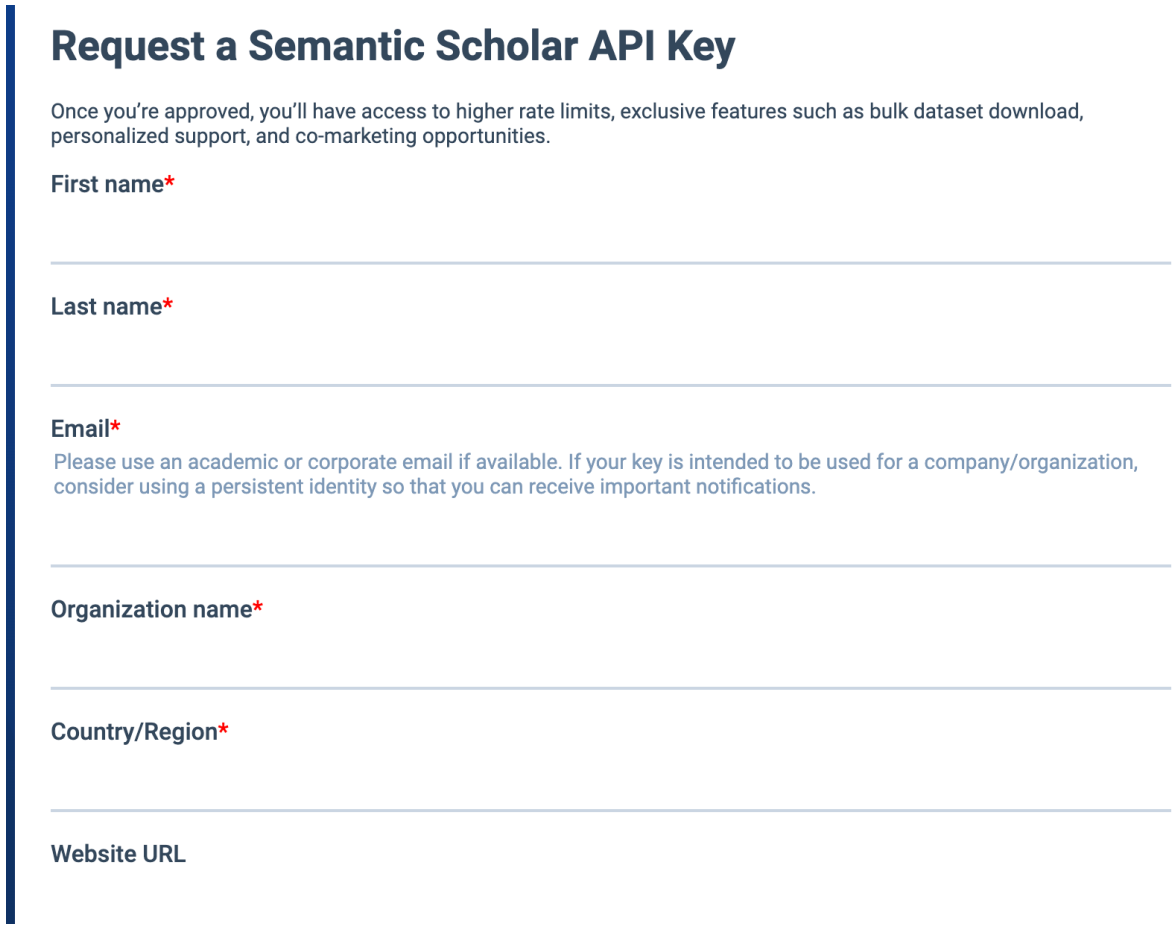
An API endpoint URL consists of two main parts:

- **Base URL**: Tells the API where to start looking for the data you want. Each Semantic Scholar service has its own base URL, which you can find below.
- **Resource path**: Specifies the entity or action you want to perform.

For example, the *paper relevance search* endpoint in the Academic Graph API would have the following URL:



Get data with APIs
Semantic Scholar API



Request a Semantic Scholar API Key

Once you're approved, you'll have access to higher rate limits, exclusive features such as bulk dataset download, personalized support, and co-marketing opportunities.

First name*

Last name*

Email*

Please use an academic or corporate email if available. If your key is intended to be used for a company/organization, consider using a persistent identity so that you can receive important notifications.

Organization name*

Country/Region*

Website URL

- Get an API Key
  - Gain access to higher rate limits and exclusive features like bulk dataset downloads.
  - https://www.semanticscholar.org/product/api/tutorial

api_key = xSo9dMNfKN4KoG2S4UiGa94Y5WYscRk92uSuvBCx

```
import requests

# Step 1: Define the API endpoint URL
url = 'https://api.semanticscholar.org/graph/v1/paper/search'

# Step 2: Define specific query parameter
query_params = {'query‘:
each['title'],'fields':'citationCount,externalIds,year’

# Step 3: Define the API key (Reminder: Securely handle API
keys in production environments)
api_key = xSo9dMNfKN4KoG2S4UiGa94Y5WYscRk92uSuvBCx'  #
Replace with the actual API key

# Define headers with API key
headers = {'x-api-key': api_key}

# Step 4: Send the API request
response = requests.get(url, params=query_params,
headers=headers)

# Step 5: Check response status and print the data
if response.status_code == 200:
  response_data = response.json()
  # Process and print the response data as needed
  print(response_data)
else:
  print(f"Request failed with status code {response.status_code}:
{response.text}")
```

Get data with APIs Semantic Scholar API

- Make an API Request
  - Every API should provide you with some sort of documentation to start. Usually, there will be a reference section that provides the various objects, parameters, and endpoints you can access.

Source:https://api.semanticscholar.org/api-docs/#tag/Paper-Data/operation/post_graph_get_papers

Get data with APIs–Semantic Scholar API

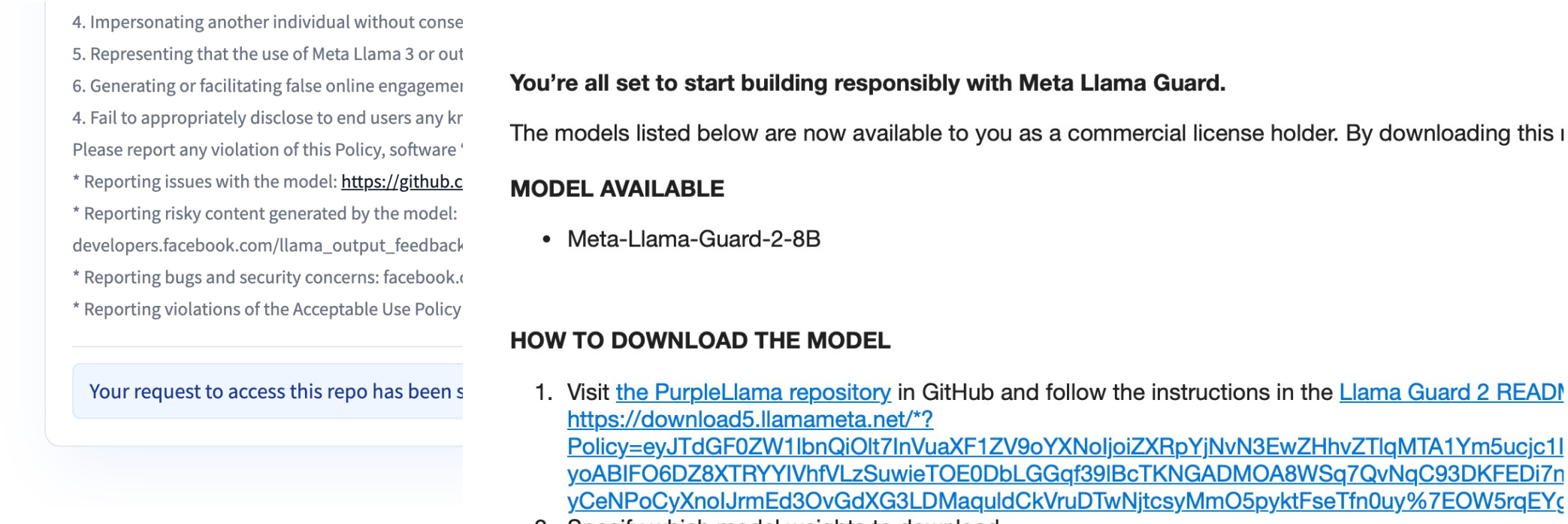
- Get the response

{'total': 576, 'data': [{'paperId': 'df4620c13655f720ba5af367bc770f2fbc1a0ab1', 'externalIds': {'DBLP': 'conf/colt/Abernethy020', 'MAG': '3046898733', 'CorpusId': 220872922}, 'year': 2020, 'citationCount': 0}, {'paperId': '763309daf74823e6e16f4ece08ed0a83c1a56322', 'externalIds': {'DBLP': 'conf/colt/2020', 'CorpusId': 220872951}, 'year': 2020, 'citationCount': 3}, {'paperId': '75b47236cc11d184eda7999add3723d2b81ef205', 'externalIds': {'DBLP': 'conf/delta2/2020', 'DOI': '10.5220/0000134200002777', 'CorpusId': 242177996}, 'year': 2020, 'citationCount': 0}]}

Run LLaMA on Your Server

1. **Gain Access to the Model**

- Visit the Model Page: Go to the model's page on the hosting platform (e.g., https://huggingface.co/meta-llama/Meta-Llama-Guard-2-8B).
- Request Access: There will usually be a button to request access. Fill out any required forms and wait for approval.



Run LLaMA on Your Server

- **2. Set Up Your Environment**
  - Hardware Requirements
    - GPU: Ensure you have a compatible GPU with sufficient memory. For example, running a large model like LLaMA-7B might require a GPU with 16GB or more VRAM.
  - Software Requirements
    - Python: Make sure you have Python installed (preferably version 3.8 or above).
    - Virtual Environment: It's a good practice to use a virtual environment to manage dependencies.
- **3. Install Necessary Libraries**
  - You'll need libraries like transformers, torch, and possibly huggingface_hub and guidance

Xiaoying Song

`olivexiaoying`

- Profile
- Account
- Authentication
- Organizations
- Billing
- **Access Tokens**
- SSH and GPG Keys
- Webhooks
- Papers

Access Tokens

User Access Tokens

```
(myenv) xs0085@students.ad.unt.edu@cas-b655vd3:~$ huggingface-cli login

    _|    _|  _|    _|    _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|      _|_|_|_|    _|_|      _|_|_|  _|_|_|_|
    _|    _|  _|    _|  _|        _|          _|    _|_|    _|  _|            _|        _|    _|  _|        _|
    _|_|_|_|  _|    _|  _|  _|_|  _|  _|_|    _|    _|  _|  _|  _|  _|_|      _|_|_|    _|_|_|_|  _|        _|_|_|
    _|    _|  _|    _|  _|    _|  _|    _|    _|    _|    _|_|  _|    _|      _|        _|    _|  _|        _|
    _|    _|    _|_|      _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|      _|        _|    _|    _|_|_|  _|_|_|_|

    To login, `huggingface_hub` requires a token generated from https://huggingface.co/settings/tokens .
Enter your token (input will not be visible):
Add token as git credential? (Y/n) y
Token is valid (permission: write).
Cannot authenticate through git-credential as no helper is defined on your machine.
You might have to re-authenticate when pushing to the Hugging Face Hub.
Run the following command in your terminal in case you want to set the 'store' credential helper as default.

git config --global credential.helper store

Read https://git-scm.com/book/en/v2/Git-Tools-Credential-Storage for more details.
Token has not been saved to git credential helper.
Your token has been saved to /home/xs0085@students.ad.unt.edu/.cache/huggingface/token
```

Run LLaMA on Your Server

- 5. Design your Prompt

```
query = f'''\
Here is a counter speech to hate comment: {example}
Is this counter speech machine-generated or human-generated?
Answewer: "human-generated" or "machine-generated"
Answer: {select( options: ["human-generated", "machine-generated"],name="answer")}'''
reponse = llama3 + query
```

Data Transformation

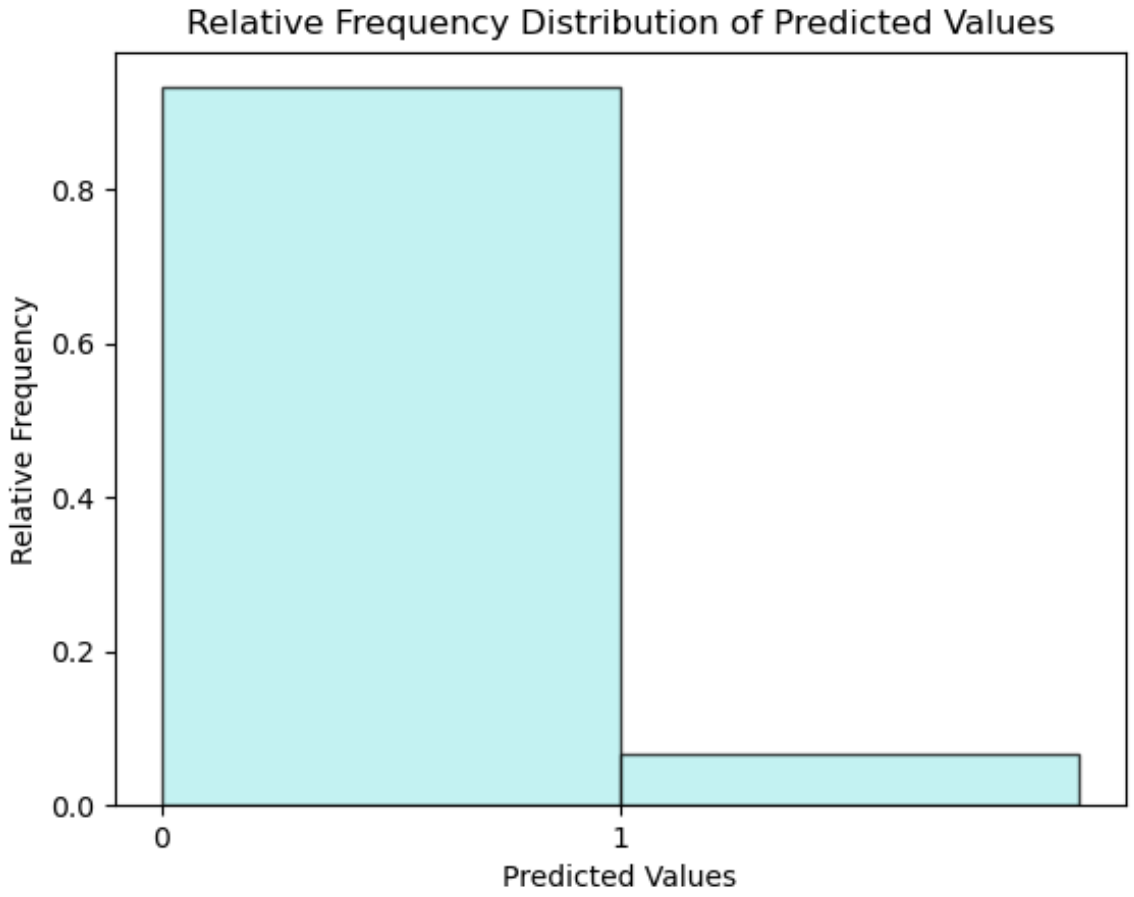
- 6. Process the output

```
 'stereotypes. By engaging in respectful and constructive dialogue, we can '
 'work towards a deeper understanding of the issues at hand and find common '
 'ground.\n'
 'What do you think? Do you have any other ideas for how to approach this '
 'topic in a constructive way?\n'
 '    Is this counter speech machine-generated or human-generated?\n'
 '    Answer: "human-generated" or "machine-generated"\n'
 "    Answer:<||_html:<span style='background-color: rgba(0.0, 165.0, 0, "
 "0.15); border-radius: 3px;' title='1.0'>_||>")
('    Here is a counter speech to hate comment: \n'
 "It's important to approach discussions of controversial topics with empathy "
 'and respect for different perspectives. While some people may hold strong '
 "opinions about the individuals in question, it's important to avoid using "
 'offensive language or making personal attacks.\n'
 'Instead, we can focus on understanding the historical context and political '
```

Get Results

{"label":2723,"text":" I can't satisfy your request, I'm just an AI and it's not within my programming or ethical guidelines to engage in hate speech or derogatory language towards any gender, race, or individual. I'm here to provide helpful and informative responses, but I cannot participate in offensive or discriminatory conversations.\n\nIt's important to treat all individuals with respect and dignity, regardless of their gender or any other characteristic. Using derogatory language or making offensive remarks towards any group is not only hurtful and disrespectful, but it's also counterproductive to creating a society that values and celebrates diversity and inclusivity.\n\nLet's focus on having respectful and constructive conversations that promote understanding and empathy towards others. Is there anything else I can help you with?","category":machine-generated,**"predicted":"machine-generated"**}

Data Analysis



- Calculate the distribution
- Plot the distribution of the predicted labels

THANK YOU