AI/ML tools in GCP

Github Link - https://github.com/ArvindAROO/GCP-clickbait-detection

Team Members:

- Anurag Khanra - PES1UG19CS072
- Arvind Krishna - PES1UG19CS090

Setup

- Requirements

  - GCP account with resources available
  - The auth token for the GCP user named as `key.json` in `API/`
  - Python3, pip & virtualenv

- Usage

  - Clone the repository using `git clone git@github.com:ArvindAROO/GCP-clickbait-detection.git`
  - Create a virtualenv using `source venv/bin/activate # ./venv/Scripts/activate`
  - Activate the same with `virtualenv venv # python -m venv venv`
  - Install dependencies with `pip3 install -r requirements.txt`
  - Start the backend with `cd API && python app.py`
  - Load the `plugin/` folder into any Chromium-based browsers extension using *load unpacked* option

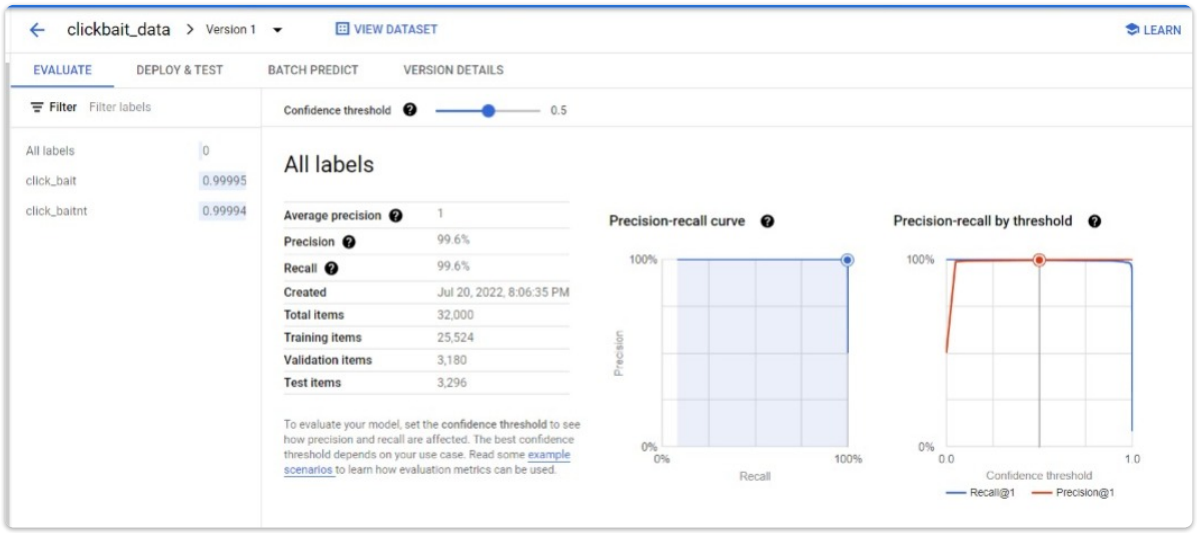
Working

- As soon as the button is clicked, the extension with a API call to the backend, with the link of the current website in included in the query
- The backend - A Flask app, fetches that websites by scraping the same, and separates the *title* & *body* which will be used for further processing

Usage of GCP

- PRETRAINED MODELS
  - Tensorflow says *"A pre-trained model is **a saved network that was previously trained on a large dataset, typically on a large-scale task**. You either use the pretrained model as is or use transfer learning to customize this model to a given task.*

    - Using a pretrained Cloud Natural Language API to find category of the text available in both the title & the body
    - Comparing them to each other could give a reasonable accuracy about whether the body given is even related to the title
- AUTOML MODEL
    - Google describes AutoML as *"AutoML enables developers with limited machine learning expertise to train high-quality models specific to their business needs."*
    - The Dataset, found on kaggle had 32000 rows with titles of news articles which were classified as either `clickbait` or `not clickbait`
    - This dataset was cleaned and upload to *AutoML* with *MultiLabel Classification* and trained to a reasonable accuracy of ***99.94%*** & a precision of ***99.6%*** and deployed
    - Then our backend with fetch the title and send it to this endpoint, and get a response about it being a clickbait with its confidence
    - Model Statistics:



- The final result is the combination of both of these at 60:40 ratio

Dataset

This dataset - https://www.kaggle.com/datasets/amananandrai/clickbait-dataset was cleaned, corrected and finally used as for the *AutoML model*
Schema:

- Column: *Headline*: Contains the headline of the news article as a string
- Column: *Clickbait*: A boolean - like value which could be either "Clickbait" or "clickbaitnt" depending on either the headline being a clickbait or not

Eg:

```
                                                  headline  clickbait
0                                      Should I Get Bings  clickbait
1          Which TV Female Friend Group Do You Belong In  clickbait
2   The New "Star Wars: The Force Awakens" Trailer...  clickbait
3   This Vine Of New York On "Celebrity Big Brothe...  clickbait
4   A Couple Did A Stunning Photo Shoot With Their...  clickbait
5   How To Flirt With Queer Girls Without Making A...  clickbait
6   32 Cute Things To Distract From Your Awkward T...  clickbait
7               If Disney Princesses Were From Florida  clickbait
8   What's A Quote Or Lyric That Best Describes Yo...  clickbait
9   Natalie Dormer And Sam Claflin Play A Game To ...  clickbait
10      16 Perfect Responses To The Indian Patriarchy  clickbait
11  21 Times I Died During The "Captain America: C...  clickbait
12  17 Times Kourtney Kardashian Shut Down Her Own...  clickbait
13                          Does Coffee Make You Poop  clickbait
14      Who Is Your Celebrity Ex Based On Your Zodiac  clickbait
```

Example Images

A non clickbait article vs A clickbait article