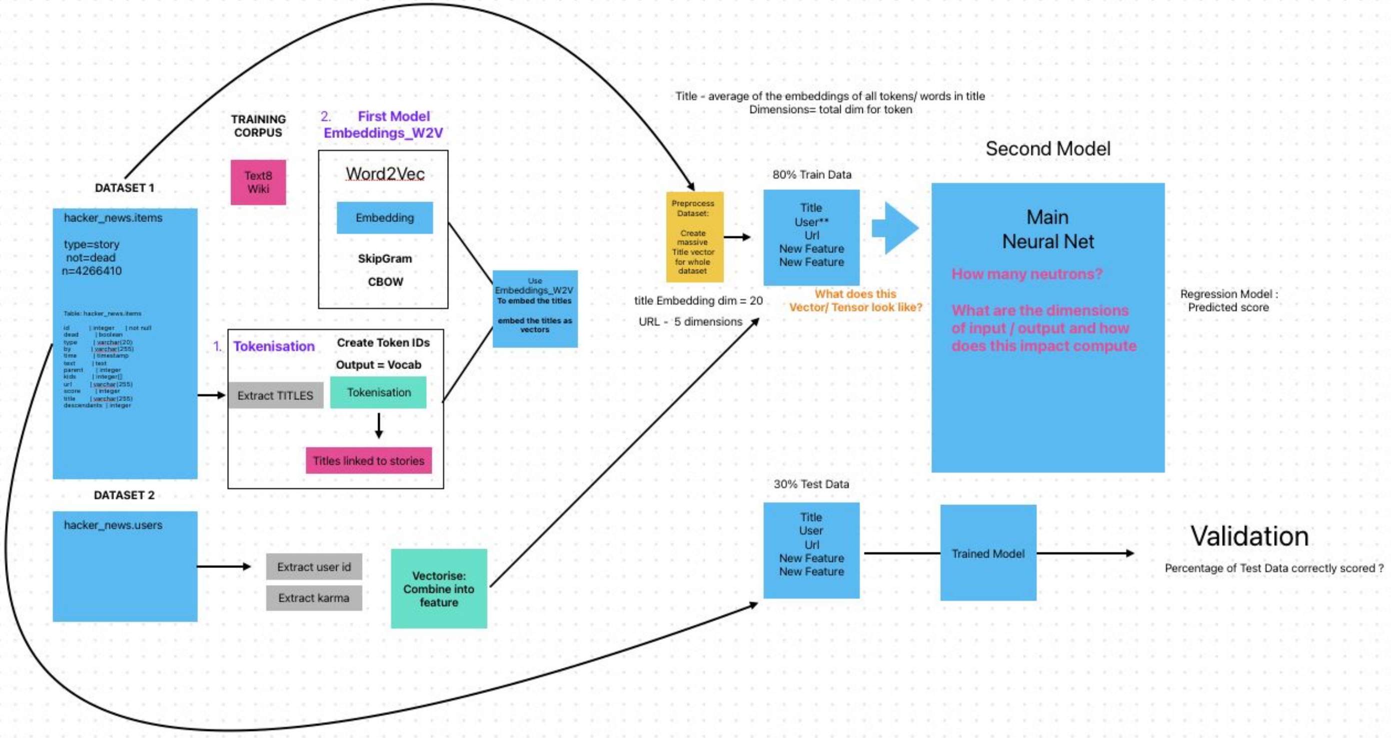
CBOW + NN

Bayesian Buccanneers
Ben, Umut, Tomas, AJ

Vibe Coding to Vibe Coping

Using Natural Language to cope.

Schema for Word2Vec, CBOW and NN



DATASET 1

hacker_news.items

type=story
not=dead
n=4266410

Table: hacker_news.items

```
id          | integer     | not null
dead        | boolean
type        | varchar(20)
by          | varchar(255)
time        | timestamp
text        | text
parent      | integer
kids        | integer[]
url         | varchar(255)
score       | integer
title       | varchar(255)
descendants | integer
```

TRAINING CORPUS

Text8
Wiki

2. First Model
Embeddings_W2V

Word2Vec

Embedding

SkipGram

CBOW

1. Tokenisation

Create Token IDs

Output = Vocab

Extract TITLES

Tokenisation

Titles linked to stories

Use
Embeddings_W2V
To embed the titles

embed the titles as vectors

DATASET 2

hacker_news.users

Extract user id

Extract karma

Vectorise:
Combine into feature

Title - average of the embeddings of all tokens/ words in title
Dimensions= total dim for token

Preprocess Dataset:

Create massive Title vector for whole dataset

title Embedding dim = 20

URL - 5 dimensions

80% Train Data

Title
User**
Url
New Feature
New Feature

What does this
Vector/ Tensor look like?

Second Model

Main
Neural Net

How many neutrons?

What are the dimensions of input / output and how does this impact compute

Regression Model :
Predicted score

30% Test Data

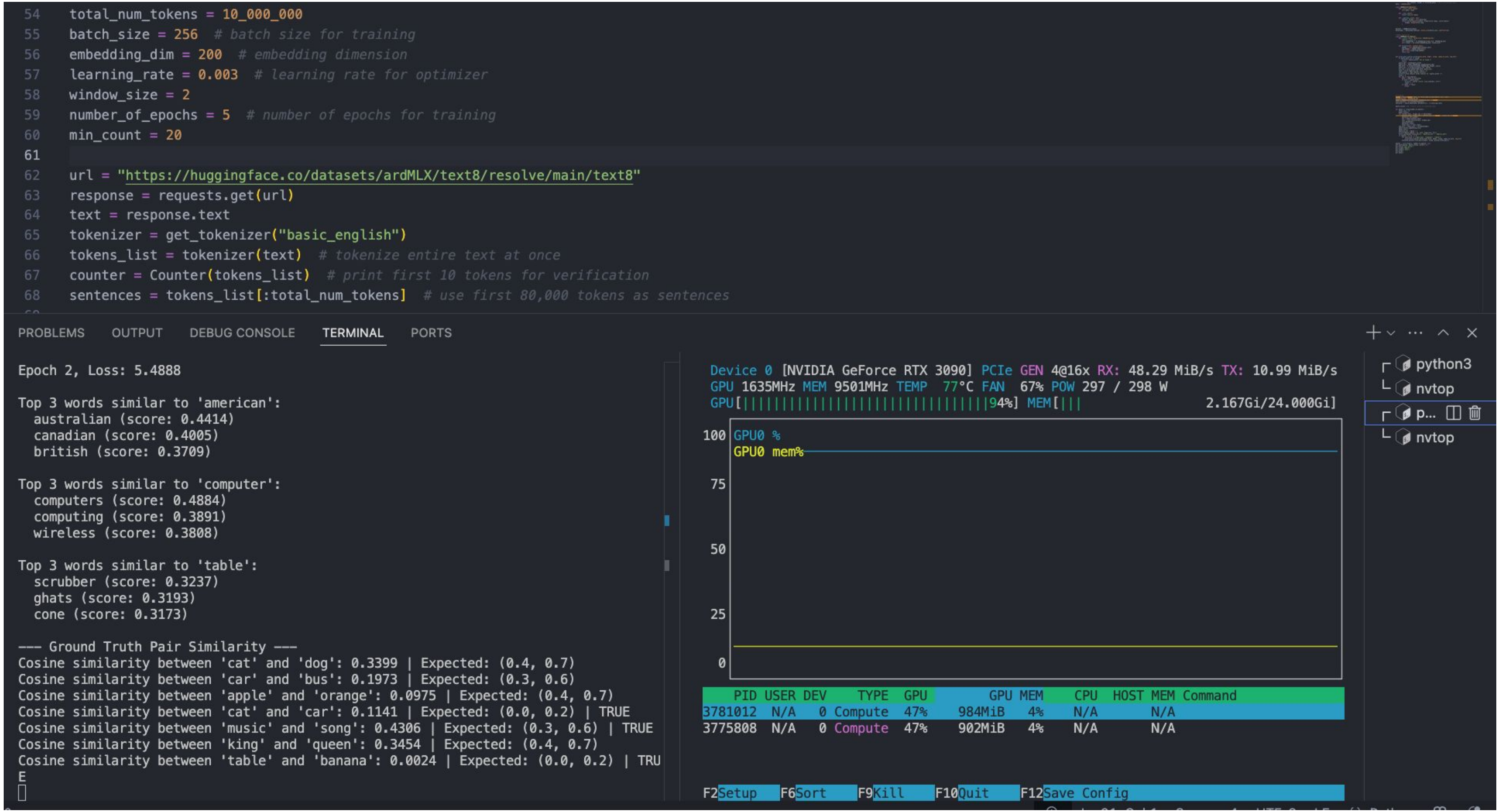
Title
User
Url
New Feature
New Feature

Trained Model

Validation

Percentage of Test Data correctly scored ?

Training CBOW via GPU



```python
total_num_tokens = 10_000_000
batch_size = 256  # batch size for training
embedding_dim = 200  # embedding dimension
learning_rate = 0.003  # learning rate for optimizer
window_size = 2
number_of_epochs = 5  # number of epochs for training
min_count = 20

url = "https://huggingface.co/datasets/ardMLX/text8/resolve/main/text8"
response = requests.get(url)
text = response.text
tokenizer = get_tokenizer("basic_english")
tokens_list = tokenizer(text)  # tokenize entire text at once
counter = Counter(tokens_list)  # print first 10 tokens for verification
sentences = tokens_list[:total_num_tokens]  # use first 80,000 tokens as sentences
```

```
Epoch 2, Loss: 5.4888

Top 3 words similar to 'american':
  australian (score: 0.4414)
  canadian (score: 0.4005)
  british (score: 0.3709)

Top 3 words similar to 'computer':
  computers (score: 0.4884)
  computing (score: 0.3891)
  wireless (score: 0.3808)

Top 3 words similar to 'table':
  scrubber (score: 0.3237)
  ghats (score: 0.3193)
  cone (score: 0.3173)

--- Ground Truth Pair Similarity ---
Cosine similarity between 'cat' and 'dog': 0.3399 | Expected: (0.4, 0.7)
Cosine similarity between 'car' and 'bus': 0.1973 | Expected: (0.3, 0.6)
Cosine similarity between 'apple' and 'orange': 0.0975 | Expected: (0.4, 0.7)
Cosine similarity between 'cat' and 'car': 0.1141 | Expected: (0.0, 0.2) | TRUE
Cosine similarity between 'music' and 'song': 0.4306 | Expected: (0.3, 0.6) | TRUE
Cosine similarity between 'king' and 'queen': 0.3454 | Expected: (0.4, 0.7)
Cosine similarity between 'table' and 'banana': 0.0024 | Expected: (0.0, 0.2) | TRU
E
```

```
Device 0 [NVIDIA GeForce RTX 3090] PCIe GEN 4@16x RX: 48.29 MiB/s TX: 10.99 MiB/s
GPU 1635MHz MEM 9501MHz TEMP  77°C FAN  67% POW 297 / 298 W
GPU[||||||||||||||||||||||||||||||94%] MEM[|||                2.167Gi/24.000Gi]

100 GPU0 %
    GPU0 mem%
 75

 50

 25

  0

   PID USER DEV     TYPE  GPU        GPU MEM    CPU  HOST MEM Command
3781012  N/A   0 Compute  47%    984MiB   4%    N/A       N/A
3775808  N/A   0 Compute  47%    902MiB   4%    N/A       N/A

F2Setup   F6Sort    F9Kill    F10Quit   F12Save Config
```

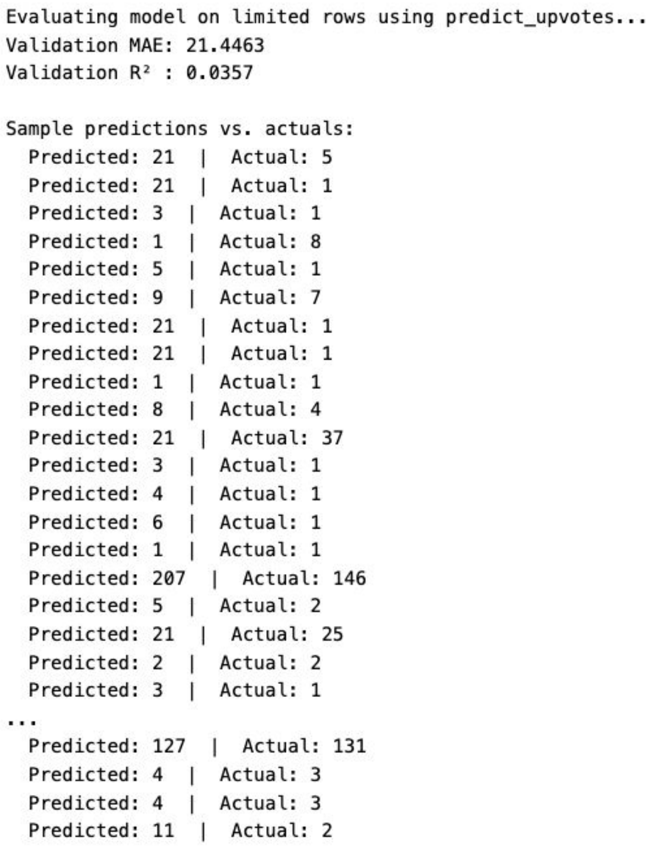
Evaluating the results

```
# Compute metrics

mae = mean_absolute_error(all_targets, all_preds)
r2  = r2_score(all_targets, all_preds)


print(f"Validation MAE: {mae:.4f}")
print(f"Validation R² : {r2:.4f}")


for p, t in list(zip(all_preds, all_targets))[:20]:
    print(f"  Predicted: {p}  |  Actual: {t}")
```

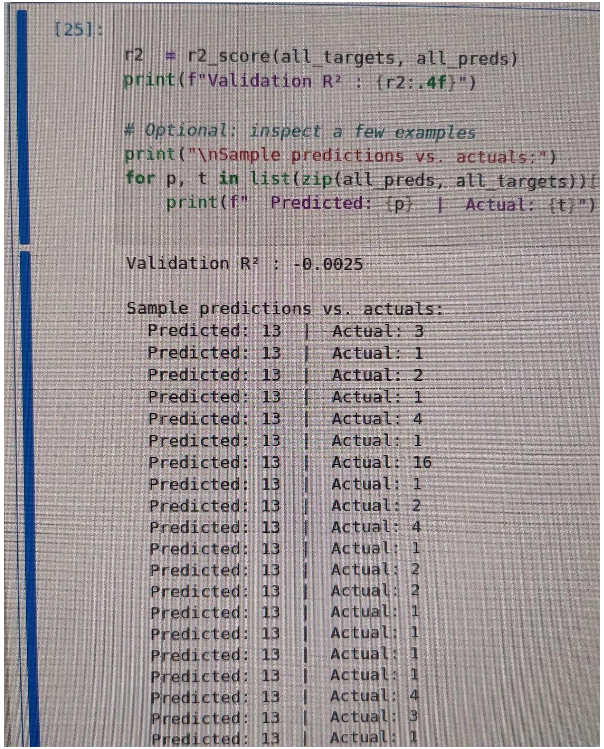


```
Evaluating model on limited rows using predict_upvotes...
Validation MAE: 21.4463
Validation R² : 0.0357

Sample predictions vs. actuals:
  Predicted: 21  |  Actual: 5
  Predicted: 21  |  Actual: 1
  Predicted: 3  |  Actual: 1
  Predicted: 1  |  Actual: 8
  Predicted: 5  |  Actual: 1
  Predicted: 9  |  Actual: 7
  Predicted: 21  |  Actual: 1
  Predicted: 21  |  Actual: 1
  Predicted: 1  |  Actual: 1
  Predicted: 8  |  Actual: 4
  Predicted: 21  |  Actual: 37
  Predicted: 3  |  Actual: 1
  Predicted: 4  |  Actual: 1
  Predicted: 6  |  Actual: 1
  Predicted: 1  |  Actual: 1
  Predicted: 207  |  Actual: 146
  Predicted: 5  |  Actual: 2
  Predicted: 21  |  Actual: 25
  Predicted: 2  |  Actual: 2
  Predicted: 3  |  Actual: 1
...
  Predicted: 127  |  Actual: 131
  Predicted: 4  |  Actual: 3
  Predicted: 4  |  Actual: 3
  Predicted: 11  |  Actual: 2
```

ML Luck 🍀

When we thought predictions can’t get worse...



```
[25]:
r2  = r2_score(all_targets, all_preds)
print(f"Validation R² : {r2:.4f}")

# Optional: inspect a few examples
print("\nSample predictions vs. actuals:")
for p, t in list(zip(all_preds, all_targets))[
    print(f"  Predicted: {p}  |  Actual: {t}")
```

```
Validation R² : -0.0025

Sample predictions vs. actuals:
  Predicted: 13  |  Actual: 3
  Predicted: 13  |  Actual: 1
  Predicted: 13  |  Actual: 2
  Predicted: 13  |  Actual: 1
  Predicted: 13  |  Actual: 4
  Predicted: 13  |  Actual: 1
  Predicted: 13  |  Actual: 16
  Predicted: 13  |  Actual: 1
  Predicted: 13  |  Actual: 2
  Predicted: 13  |  Actual: 4
  Predicted: 13  |  Actual: 1
  Predicted: 13  |  Actual: 2
  Predicted: 13  |  Actual: 2
  Predicted: 13  |  Actual: 1
  Predicted: 13  |  Actual: 1
  Predicted: 13  |  Actual: 1
  Predicted: 13  |  Actual: 1
  Predicted: 13  |  Actual: 4
  Predicted: 13  |  Actual: 3
  Predicted: 13  |  Actual: 1
```

Streamlit UI

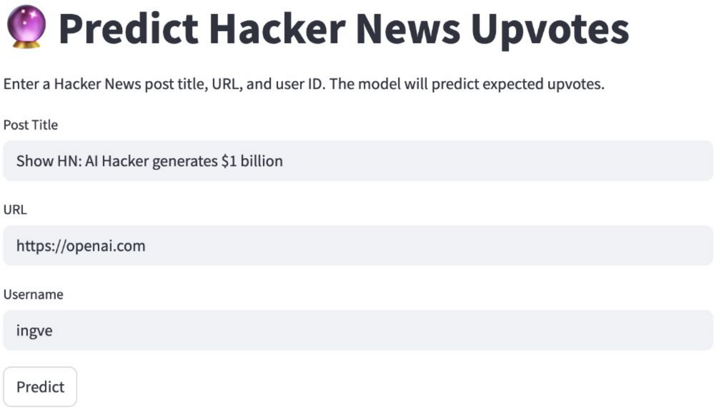
New Venv

Requirements.txt

Upload the .py code

Upload the embeddings

Run...



🔮 Predict Hacker News Upvotes

Enter a Hacker News post title, URL, and user ID. The model will predict expected upvotes.

Post Title: Show HN: AI Hacker generates $1 billion

URL: https://openai.com

Username: ingve

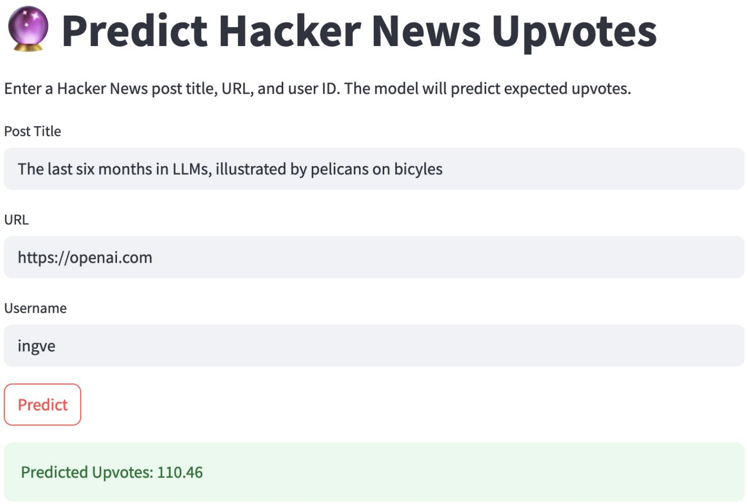
Learning Points

1. Aj: Utilising SQL queries to speed up data parsing - Setting up evaluation/validation as early as possible

2. Tomas: Training monitoring + understanding, then scaling up

3. Umut: Different CBOW embeddings, Feature engineering. Keeping it simple works best. Learning Rate makes big difference... Getting a simple model to work is and then tweaking is the best way to progress...

4. Ben: sshing into remote GPU, playing with the parameters in word2vec & understanding relationships.



🔮 Predict Hacker News Upvotes

Enter a Hacker News post title, URL, and user ID. The model will predict expected upvotes.

Post Title

The last six months in LLMs, illustrated by pelicans on bicyles

URL

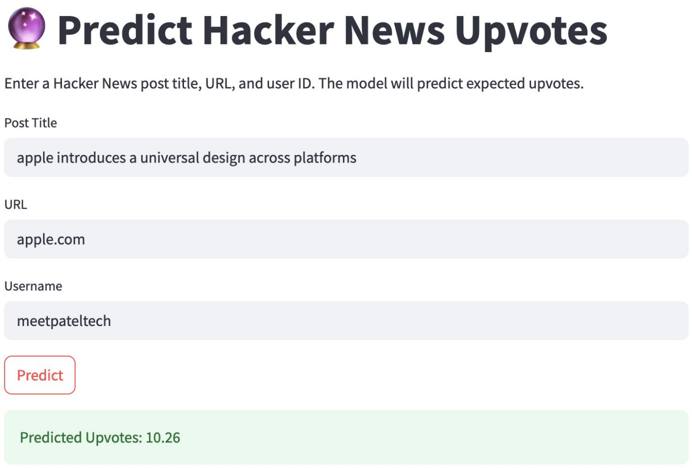
https://openai.com

Username

ingve

Predicted Upvotes: 110.46

Real Upvotes: 942



🔮 Predict Hacker News Upvotes

Enter a Hacker News post title, URL, and user ID. The model will predict expected upvotes.

Post Title

apple introduces a universal design across platforms

URL

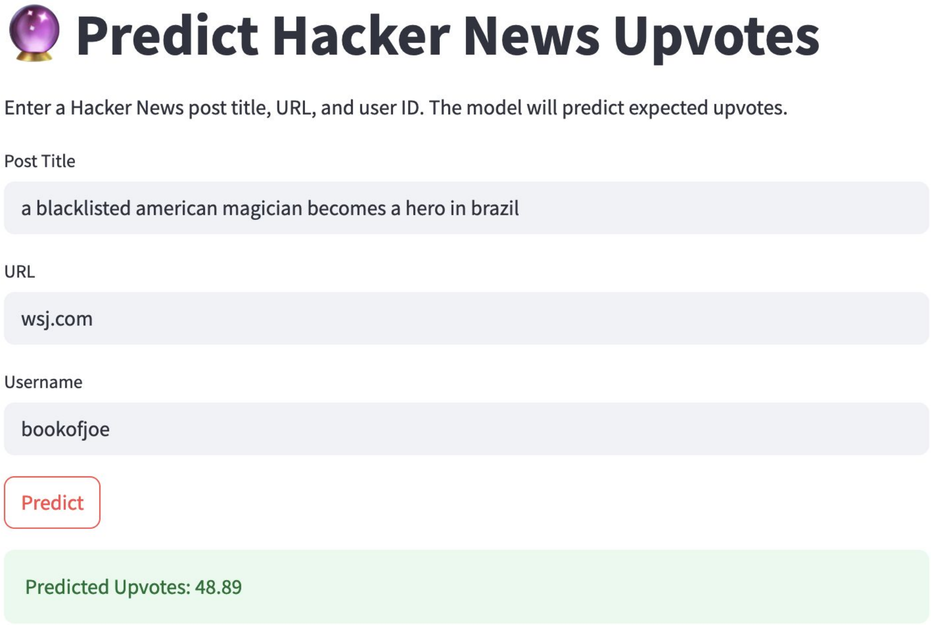
apple.com

Username

meetpateltech

Predicted Upvotes: 10.26

Real Upvotes: 727



🔮 Predict Hacker News Upvotes

Enter a Hacker News post title, URL, and user ID. The model will predict expected upvotes.

Post Title

a blacklisted american magician becomes a hero in brazil

URL

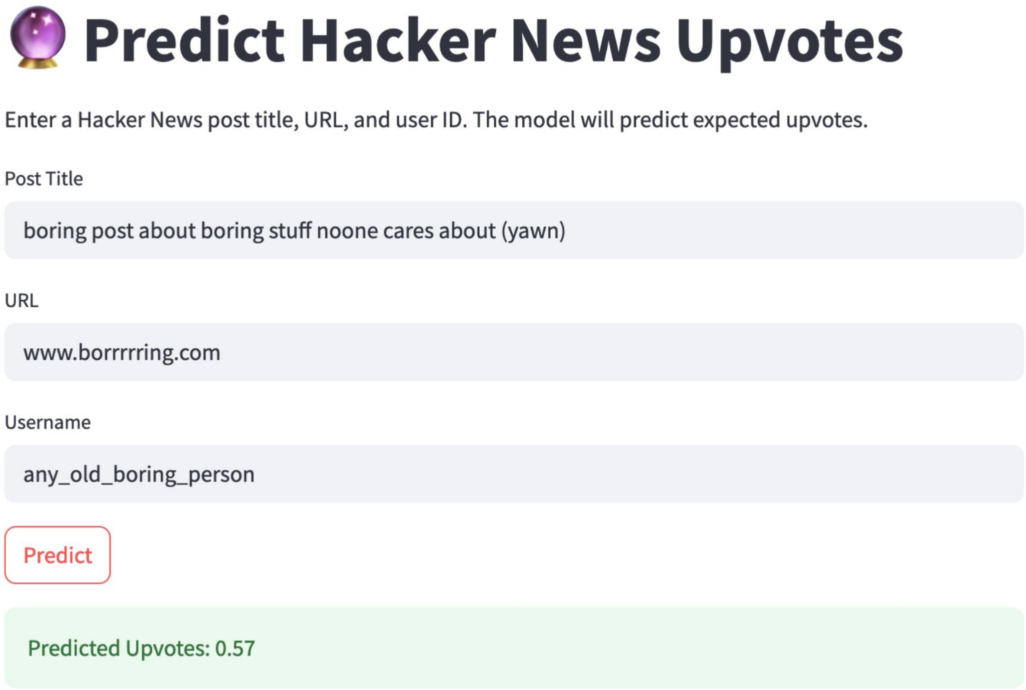
wsj.com

Username

bookofjoe

Predicted Upvotes: 48.89

Real Upvotes: 118



🔮 Predict Hacker News Upvotes

Enter a Hacker News post title, URL, and user ID. The model will predict expected upvotes.

Post Title

boring post about boring stuff noone cares about (yawn)

URL

www.borrrrring.com

Username

any_old_boring_person

Predicted Upvotes: 0.57

Real Upvotes: 0 (We guess!)