Comic Book Recommendation System

By: Kelly Dong

Project Overview

The comic book industry is a behemoth mess with an enormous amount of volumes and issues. It is a constant challenge to search through the large backlog for a story that will interest you.

Using natural language processing and feature engineering, we will create a model that takes in an issue name that you have read before and output recommendations that are similar to your input.

The resulting product will save the user time and energy as they browse for their next comic book.

Dataset Overview & Preprocessing

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1900 entries, 0 to 1899
Data columns (total 23 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   index              1900 non-null   int64
 1   api_detail_url     1900 non-null   object
 2   associated_images  1900 non-null   object
 3   character_credits  1900 non-null   object
 4   character_died_in  1900 non-null   object
 5   concept_credits    1900 non-null   object
 6   cover_date         1900 non-null   datetime64[ns]
 7   date_added         1900 non-null   datetime64[ns]
 8   date_last_updated  1900 non-null   datetime64[ns]
 9   description        1900 non-null   object
 10  has_staff_review   1900 non-null   object
 11  id                 1900 non-null   int64
 12  image              1900 non-null   object
 13  issue_number       1900 non-null   object
 14  location_credits   1900 non-null   object
 15  name               1900 non-null   object
 16  object_credits     1900 non-null   object
 17  person_credits     1900 non-null   object
 18  site_detail_url    1900 non-null   object
 19  story_arc_credits  1900 non-null   object
 20  team_credits       1900 non-null   object
 21  team_disbanded_in  1900 non-null   object
 22  volume             1900 non-null   object
dtypes: datetime64[ns](3), int64(2), object(18)
```

```
def clean_description(html_text):
    soup = BeautifulSoup(html_text, 'html.parser')
    cleaned_text = soup.get_text(separator='\n')
    return cleaned_text

def get_names(json_str, index):
    json_str = json_str.replace("'",'"')
    try:
        lst = json.loads(json_str)
    except json.JSONDecodeError as e:
        print(f"JSONDecodeError at row {index}: {e}")
        return []
    names = [item["name"] for item in lst]
    return names
```

```
#
```

```
def to_string(lst):
    string = ""
    for item in lst:
        string = string + " " + item
    return string
```

```
clean_combined["character_credits"] = clean_combined["character_credits"].apply(to_string)
clean_combined["location_credits"] = clean_combined["location_credits"].apply(to_string)
clean_combined["person_credits"] = clean_combined["person_credits"].apply(to_string)
```

```
clean_combined["combined"] = clean_combined["description"] + clean_combined["character_credits"] + clean_combined["location_credits"] + clean_combined["person_credits"]
```

```
clean_combined["combined"][0]
```

```
'Cover by J. Winslow Mortimer.\n"\nThe Crazy Crime Clown!\n" written by Alvin Schwartz, penciled by Dick Sprang and inked by Charles Paris.\nCasey the Cop\n"\nThe Movie That Killed Batman\n" written by Bill Finger, penciled and inked by Dick Spran
g.\nLittle Pete\nVarsity Vic\n"\nThe Water Crimes of Mr.Hydro\n" written by Bill Finger, penciled by Lew Schwartz and inked by Stan Kaye.\nJerry the Jitterbug Alfred Pennyworth Batman Casey the Cop Dick Grayson James Gordon Jerry the Jitterbug Joke
r Little Pete Varsity Vic Batcave Gotham City Gotham State Penitentiary Alvin Schwartz Bill Finger Bob Kane Charles Paris Dick Sprang Lew Sayre Schwartz Pat Gordon Winslow Mortimer'
```

Model & Evaluation

BASIC CONTENT-BASED RECOMMENDATION MODEL

```
]: tfidf = TfidfVectorizer(stop_words='english')
```

```
]: tfidf_matrix = tfidf.fit_transform(clean_combined['combined'])
```

```
]: cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)
```

```
]: def get_recommendations(df, title, cosine_sim=cosine_sim):
    # Get the index of the issue that matches the title
    idx = df[df['name'] == title].index[0]
    # Get the pairwise similarity scores of all issues with that issue
    sim_scores = list(enumerate(cosine_sim[idx]))
    # Sort the issues based on the similarity scores
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    # Get the scores of the 10 most similar issues
    sim_scores = sim_scores[1:11]
    # Get the issue indices
    issue_indices = [i[0] for i in sim_scores]
    # Return the top 10 most similar issues
    return df[['name', 'issue_number', 'description']].iloc[issue_indices]
```

Model & Evaluation (just description)

```
recommendations = get_recommendations(clean_combined, "Welcome to Gotham Academy")
recommendations
```

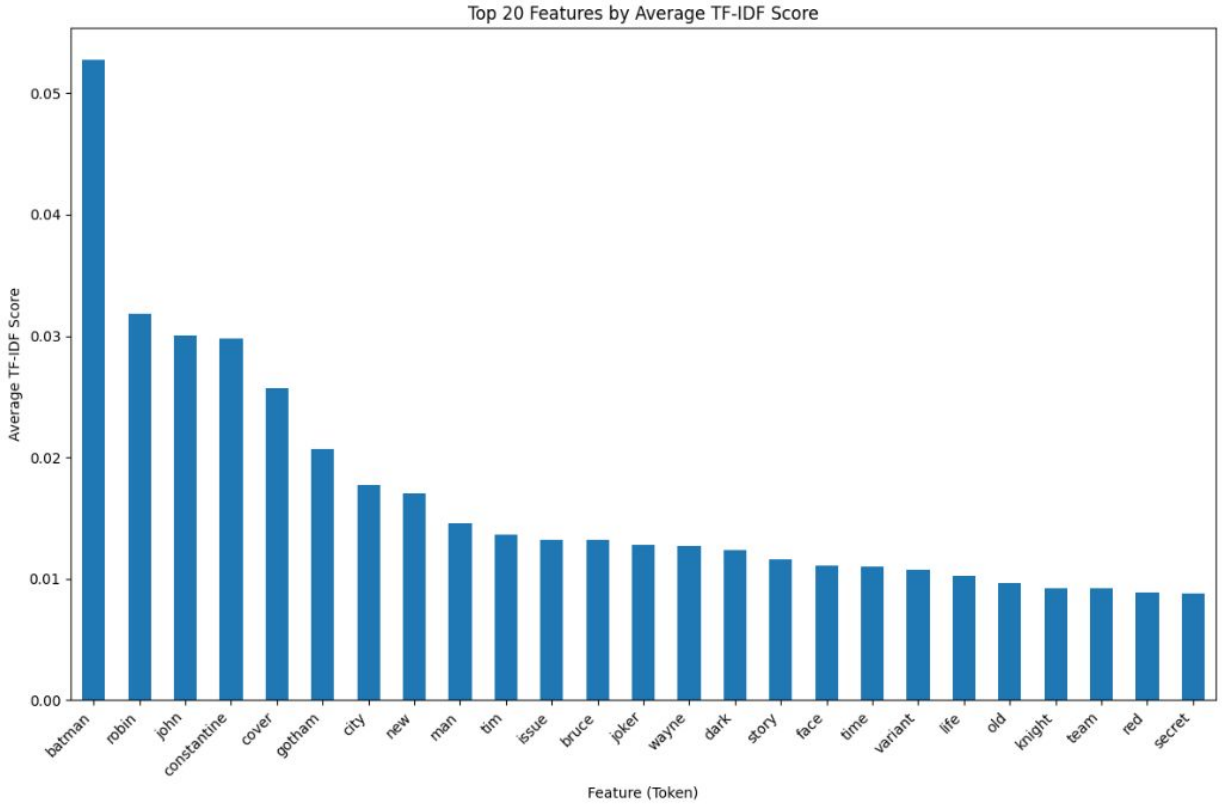
	name	issue_number	description
1025	Yearbook Part One; Animal Science 101; Queen G...	14	An all-new era of GOTHAM ACADEMY begins here w...
1016	The Secret of the Symbol	4	The hunt for the Ghost of Gotham Academy begins!
1029	Yearbook Part Five; Whatever Happened to Profe...	18	As the "Gotham Academy Yearbook" storyline com...
1017	Save The Last Dance	5	This month's assignment: Uncover the hideous s...
1023	Mission: Gotham	11	The gang is going downtown! Olive and Maps use...
1021	Calamity	9	If the gang thought it was hard to keep up wit...
⇨ **723**	Attack on Wayne Manor; The Call	8.0	Broken and beaten, Bruce Wayne has retreated f...
1014	The Diary of Millie Jane Cobblepot	2	Olive joins the creepy Order of the Bat as an ...
1024	Robins vs. Zombies: Robin War	13	A "Robin War" tie-in! With Robins fighting cop...
⇨ **253**	Shadow Play	348	"SHADOW PLAY!" Batman and Robin return to Wayn...

Model & Evaluation (with combined)

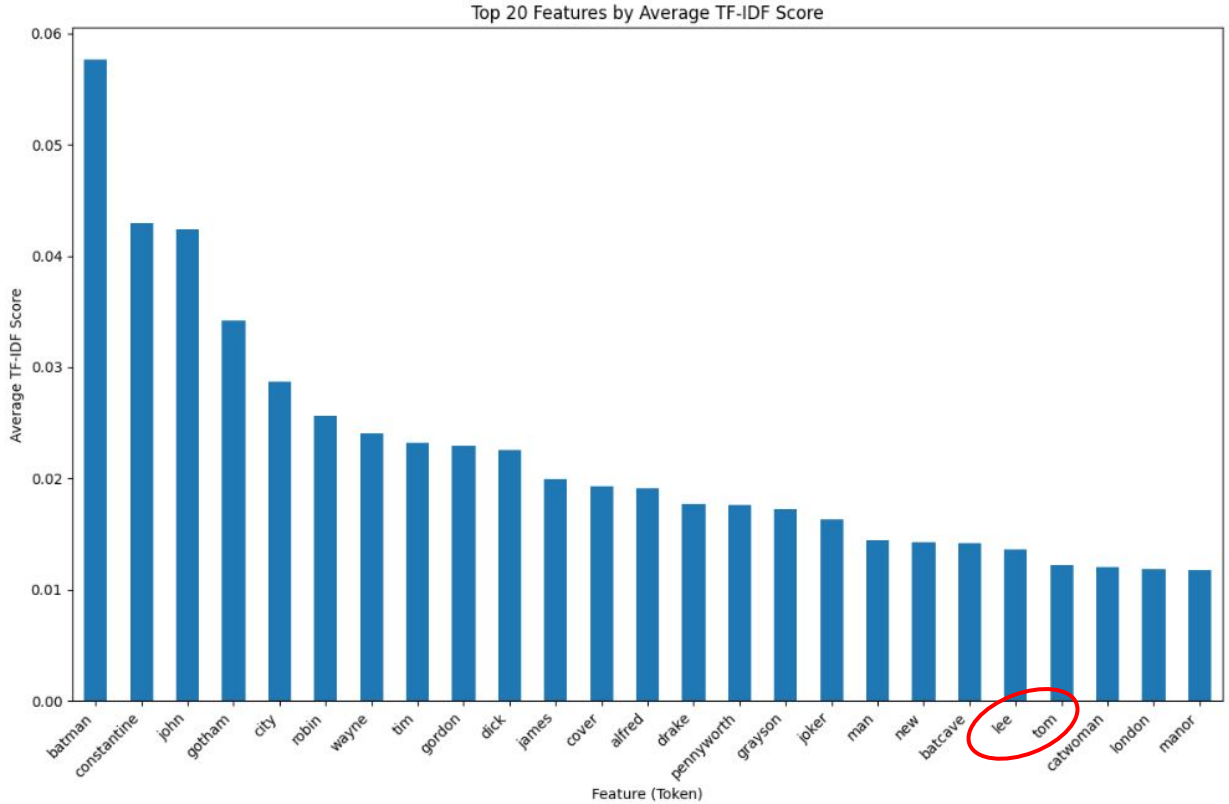
```
recommendations = get_recommendations(clean_combined, "Welcome to Gotham Academy")
recommendations
```

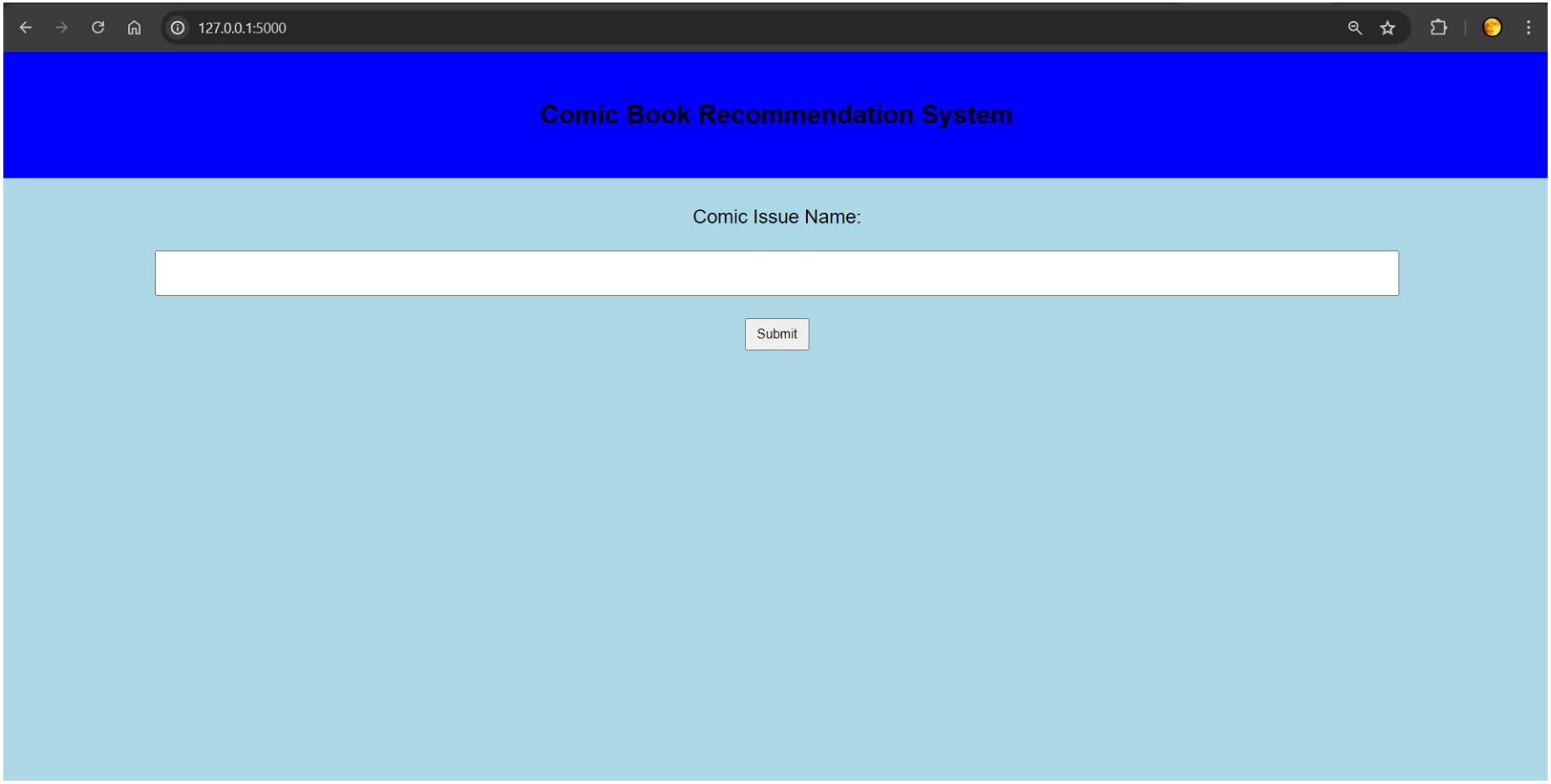
	name	issue_number	description
1017	Save The Last Dance	5	This month's assignment: Uncover the hideous s...
1016	The Secret of the Symbol	4	The hunt for the Ghost of Gotham Academy begins!
1015	The Ghost in the North Hall	3	If you thought getting detention was a pain, j...
1014	The Diary of Millie Jane Cobblepot	2	Olive joins the creepy Order of the Bat as an ...
1025	Yearbook Part One; Animal Science 101; Queen G...	14	An all-new era of GOTHAM ACADEMY begins here w...
1018	Pizza Club	6	Holy cow, it's Taco Tuesday! Oh...and Olive batt...
1029	Yearbook Part Five; Whatever Happened to Profe...	18	As the "Gotham Academy Yearbook" storyline com...
1021	Calamity	9	If the gang thought it was hard to keep up wit...
1023	Mission: Gotham	11	The gang is going downtown! Olive and Maps use...
1026	Yearbook Part Two; Staff Party; Serpents & Sec...	15	It's part two of "Gotham Academy Yearbook"! Th...

Model & Evaluation (just description)



Model & Evaluation (with combined)




127.0.0.1:5000
Comic Book Recommendation System
Comic Issue Name:
Submit

```python
from flask import Flask, render_template

app = Flask(__name__)

@app.route("/")
def home():
    #return "Hello World!"
    return render_template("home.html")

if __name__ == "__main__":
    app.run(debug=True)

```

```html
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <title> Comic Book Recommendation System </title>
    <link rel= "stylesheet" type= "text/css" href= "{{ url_for('static',filename='styles/my
  </head>

  <body>
    <h1 id="home_title"> Comic Book Recommendation System </h1>
    <p>  </p>
    <br>
    <form id="form" action="get">
          <label for="issue_name"> Comic Issue Name: </label><br><br>
          <input type="text" id="issue_name" name="issue_name"><br><br>
          <input type="submit" id="submit_button" value="Submit">
    </form>
  </body>


</html>

```

```css
body {
    font-family: Arial, Helvetica, sans-serif;
    background-color: lightblue;
    padding: 0;
    margin: 0;
}

#home_title {
    text-align: center;
    background-color: blue;
    border-style: none;
    padding: 60px 30px 60px 30px;
    margin: 00px;
    width: max-width;
}

#form {
    text-align: center;
    font-size:24px;

}
#issue_name {
    width: 80%;
    height: 50px;
    font-size:24px;
    text-indent:20px;
}

#submit_button {
    width: 80px;
    height: 40px;
    font-size: 16px;
}
```

Last Push…

- Continue to gather data for our model
- Continue to fine tune our model
- Add our other features into the NLP model as tokens instead of combining it with the description
  - Focus on Names (“First Last”: “First”, “Last”)
- Create custom tokens for these features.
- Embed my recommendation model and function into the Web Application