

```
In [1]: import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel
import plotly.express as px
import plotly.graph_objects as go
df = pd.read_csv("train_books_data.csv")
print(df.head())
```

```

      bookID                                     title \
0         1  Harry Potter and the Half-Blood Prince (Harry ...
1         2  Harry Potter and the Order of the Phoenix (Har...
2         4  Harry Potter and the Chamber of Secrets (Harry...
3         5  Harry Potter and the Prisoner of Azkaban (Harr...
4         8  Harry Potter Boxed Set Books 1-5 (Harry Potte...

      authors average_rating
0  J.K. Rowling/Mary GrandPrÃ©  4.57
1  J.K. Rowling/Mary GrandPrÃ©  4.49
2                J.K. Rowling  4.42
3  J.K. Rowling/Mary GrandPrÃ©  4.56
4  J.K. Rowling/Mary GrandPrÃ©  4.78
```

```
In [2]: df.tail()
```

```
Out[2]:
```

	bookID		title	authors	average_rating
<b>8896</b>	34460		Quicksand	Jun'ichirÅ Tanizaki/Howard Hibbett	3.65
<b>8897</b>	34462		Naomi	Jun'ichirÅ Tanizaki/Anthony H. Chambers	3.69
<b>8898</b>	34463	Seven Japanese Tales		Jun'ichirÅ Tanizaki/Howard Hibbett	3.85
<b>8899</b>	34468	A Cat a Man and Two Women		Jun'ichirÅ Tanizaki/Paul McCarthy	3.8
<b>8900</b>	34472	The Reed Cutter & Captain Shigemoto's Mother		Jun'ichirÅ Tanizaki/Anthony H. Chambers	3.67

```
In [3]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8901 entries, 0 to 8900
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  -
0   bookID          8901 non-null   int64
1   title           8901 non-null   object
2   authors         8901 non-null   object
3   average_rating  8901 non-null   object
dtypes: int64(1), object(3)
memory usage: 278.3+ KB
```

```
In [4]: df.describe()
```

Out[4]:

	bookID
count	8901.000000
mean	16578.236266
std	9985.768427
min	1.000000
25%	7966.000000
50%	16006.000000
75%	25045.000000
max	34472.000000

In [5]:

```
y=df[['bookID','title','authors','average_rating']]
y
```

Out[5]:

	bookID	title	authors	average_rating
0	1	Harry Potter and the Half-Blood Prince (Harry ...	J.K. Rowling/Mary GrandPrÃ©	4.57
1	2	Harry Potter and the Order of the Phoenix (Har...	J.K. Rowling/Mary GrandPrÃ©	4.49
2	4	Harry Potter and the Chamber of Secrets (Harry...	J.K. Rowling	4.42
3	5	Harry Potter and the Prisoner of Azkaban (Harr...	J.K. Rowling/Mary GrandPrÃ©	4.56
4	8	Harry Potter Boxed Set Books 1-5 (Harry Potte...	J.K. Rowling/Mary GrandPrÃ©	4.78
...	...	...	...	...
8896	34460	Quicksand	Jun'ichirÅ Tanizaki/Howard Hibbett	3.65
8897	34462	Naomi	Jun'ichirÅ Tanizaki/Anthony H. Chambers	3.69
8898	34463	Seven Japanese Tales	Jun'ichirÅ Tanizaki/Howard Hibbett	3.85
8899	34468	A Cat a Man and Two Women	Jun'ichirÅ Tanizaki/Paul McCarthy	3.8
8900	34472	The Reed Cutter & Captain Shigemoto's Mother	Jun'ichirÅ Tanizaki/Anthony H. Chambers	3.67

8901 rows × 4 columns

In [6]:

```
df.dtypes
```

```
Out[6]: bookID          int64
        title          object
        authors        object
        average_rating  object
        dtype: object
```

```
In [7]: df.isnull()
```

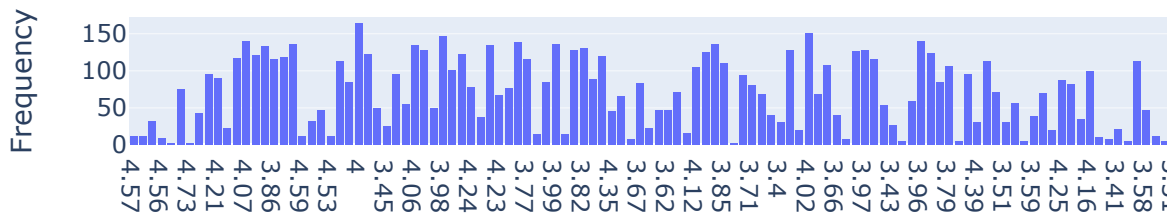
```
Out[7]:
```

	bookID	title	authors	average_rating
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False
...	...	...	...	...
8896	False	False	False	False
8897	False	False	False	False
8898	False	False	False	False
8899	False	False	False	False
8900	False	False	False	False

8901 rows × 4 columns

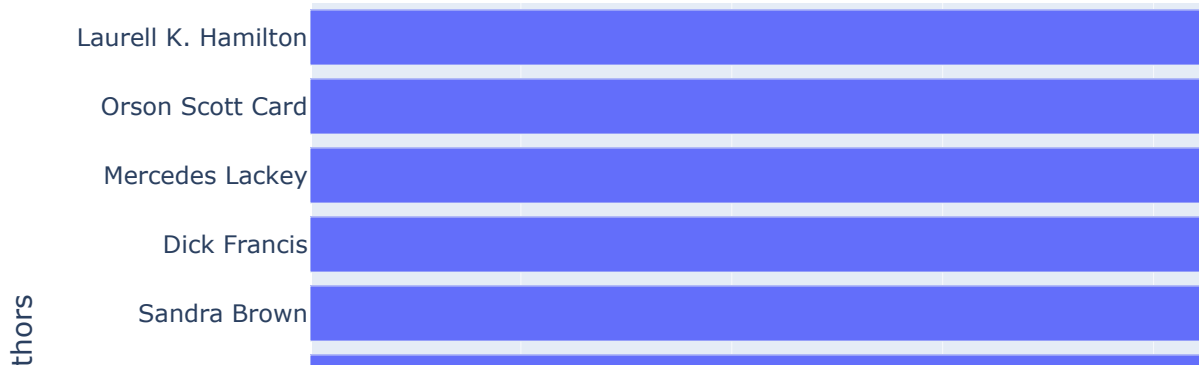
```
In [8]: fig = px.histogram(df, x='average_rating', nbins=30, title='Distribution of Average Rating')
        #px.histogram is a plotly.express function used to make histogram
        fig.update_xaxes(title_text='Average Rating')
        fig.update_yaxes(title_text='Frequency')
        fig.show()
```

## Distribution of Average Ratings



```
In [9]: top_authors = df['authors'].value_counts().head(10)
fig = px.bar(top_authors, x=top_authors.values, y=top_authors.index, orientation='h',
             labels={'x': 'Number of Books', 'y': 'Author'},
             title='Number of Books per Author')
fig.show()
```

## Number of Books per Author



```
In [10]: # Converting 'average_rating' to a numeric data type
df['average_rating'] = pd.to_numeric(df['average_rating'],
                                     errors='coerce')
```

```
In [11]: # Creating a new column 'book_content' by combining 'title' and 'authors'
df['book_content'] = df['title'] + ' ' + df['authors']
```

```
In [12]: tfidf_vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf_vectorizer.fit_transform(df['book_content'])
```

```
In [13]: # Computing the cosine similarity between books
cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)
```

```
In [14]: def recommend_books(book_title, cosine_sim=cosine_sim):
# Getting the index of the book that matches the title
idx = df[df['title'] == book_title].index[0]

# Getting the cosine similarity scores for all books with this book
sim_scores = list(enumerate(cosine_sim[idx]))

# Sorting the books based on the similarity scores
sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)

# Getting the top 10 most similar books (excluding the input book)
sim_scores = sim_scores[1:11]
```

```

# Getting the book indices
book_indices = [i[0] for i in sim_scores]

# Top 10 recommended books
return df['title'].iloc[book_indices]

```

```

In [15]: def mean_reciprocal_rank(true_book_title, recommended_books):
    for i, book_title in enumerate(recommended_books):
        if book_title == true_book_title:
            return 1 / (i + 1)
    return 0
# Returns 0 if the true book is not in the recommended list

true_book_title = "CliffsNotes on Joyce's Dubliners (Cliffs Notes)"
recommended_books_example = recommend_books(true_book_title)

if not recommended_books_example.empty:
    # Evaluating Metric: Mean Reciprocal Rank for the recommendations
    mrr_example = mean_reciprocal_rank(true_book_title, recommended_books_example)

    print(f"Recommended Books: {recommended_books_example}")
    print(f"Mean Reciprocal Rank: {mrr_example}")

```

```

Recommended Books: 4778    CliffsNotes on Faulkner's The Sound and the Fu...
3544    CliffsNotes on Faulkner's As I Lay Dying (Clif...
6189                Dubliners: Text Criticism and Notes
5367                Cliffs Notes on Voltaire's Candide
1340    Cliffs Notes on Fitzgerald's the Great Gatsby
5595                Cliffs Notes on Wright's Black Boy
214      Golding's Lord of the Flies (Cliffs Notes)
1607    Cliffs notes on Warren's All the King's Men
612                                           J.K.Rowling
2432    Cliffs Notes on Shakespeare's Macbeth
Name: title, dtype: object
Mean Reciprocal Rank: 0

```