Project Based Learning Report

On

BOOK RECOMMENDATION SYSTEM

Submitted in the partial fulfillment of the requirements

For the Project based learning in **Essentials of Data Science**

In

Electronics & Communication Engineering

By

2014111082	Kartik Mudgal
2014111093	Hritik Raj
2014111095	Vansh Raj

Under the guidance of Course In – charge

Prof. D S Lavhkare

Department of Electronics & Communication Engineering

Bharati Vidyapeeth
(Deemed to be University)
College of Engineering,
Pune – 411043

ACADEMIC YEAR: 2021-22

Bharati Vidyapeeth

(Deemed to be University)

College of Engineering,

Pune – 411043

Department of Electronics & Communication Engineering,

CERTIFICATE

Certified that the Project based Learning report entitled **"BOOK RECOMMENDATION SYSTEM "** work done by

2014111082	Kartik Mudgal
2014111093	Hritik Raj
2014111095	Vansh Raj

In partial fulfillment of the requirements for the award of credits for Project Based Learning (PBL) in "**Essentials of Data Science"** of Bachelor of Technology Semester 4, in Branch name.

Date: 23 May 2022

Prof D S Lavhkare

Course In-Charge

Dr . Tanuja Dhope

PBL Co-Ordinator

Dr. Arundhati A. Shinde

Professor & Head

ELECTRONICS & COMMUNICATION ENGINEERING

Index: -

Page No.	Contents
1-1	Problem Statement with Solution
2-2	Description about project
3-3	Software Used
4-13	Code & Analysis
14-14	Conclusion & Outcome

Problem Statement :-

What is Data Science? Why learn Data Science?

Solution :-

Data science is the domain of study that deals with vast volumes of data using modern tools and techniques to find unseen patterns, derive meaningful information, and make business decisions. Data science uses complex machine learning algorithms to build predictive models. The data used for analysis can come from many different sources and presented in various formats.

Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data. Data science practitioners apply machine learning algorithms to numbers, text, images, video, audio, and more to produce artificial intelligence (AI) systems to perform tasks that ordinarily require human intelligence. In turn, these systems generate insights which analysts and business users can translate into tangible business value.

Reasons to learn Data Science are: -

1) Learning about data science provides an opportunity for you to recreate yourself.
2) We live in a digital world, everything is data-driven. There is data science in business, accounting, education, science, engineering, healthcare, technology, energy sector, government, and so on.
3) Data science is also a very promising field with lots of high paying job opportunities.
4) Basic data science skills are important for personal use.
5) Great potential to branch out with different options.
6) Become a decision-maker, not every job opportunity will give you the power to make informed business decisions. For a data scientist, that is the core responsibility.
7) Less competitive because it is a highly analytical role, competition is less, but demand is not. With a limited talent pool, there is always a challenge for businesses to hire in these roles.

Description about
Book Recommendation System

A book recommendation system is a type of recommendation system where we have to recommend similar books to the reader based on his interest. The books recommendation system is used by online websites which provide eBooks like google play books, open library, good Read's, etc. The Book Recommendation System is widely implemented using search engines comprising of data sets. The online book recommendation system involves various techniques forproviding effective suggestion for the buyers. The association mining,collaborative filtering and content filtering are the three widely employed methods for strong impact using search engines. The content-based filtering system is one in which the recommendation to the buyers are provided based on the items they have searched for.

Approach for recommendation System:

We do not want to find a similarity between users or books. we want to do that If there is user A who has read and liked x and y books, and user B has also liked this two books and now user A has read and liked some z book which is not read by B so we have to recommend z book to user B. This is what collaborative filtering is. So, this is achieved using Matrix Factorization, we will create one matrix where columns will be users and indexes will be books and value will be rating. Like we have to create a Pivot table.

Software Used: -

Jupyter Notebook: -

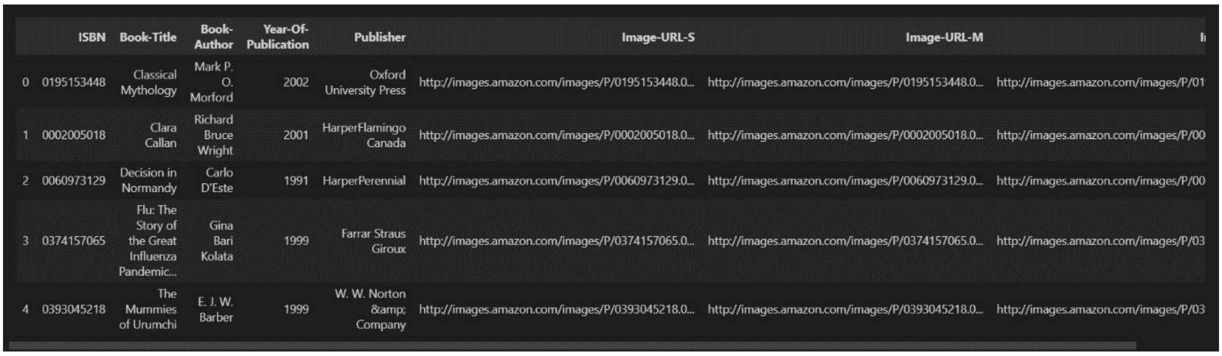
Jupyter Notebook (formerly Python Notebooks) is a web-based interactive computational environment for creating notebook documents. A Jupyter Notebook document is a browser-based REPL containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media. Underneath the interface, a notebook is a JSON document, following a versioned schema, usually ending with the “. ipynb" extension. Jupyter notebooks are built upon a number of popular open-source libraries.

Jupyter Notebook can connect to many *kernels* to allow programming in different languages. A Jupyter kernel is a program responsible for handling various types of requests (code execution, code completions, inspection), and providing a reply. Kernels talk to the other components of Jupyter using ZeroMQ, and thus can be on the same or remote machines. Unlike many other Notebook-like interfaces, in Jupyter, kernels are not aware that they are attached to a specific document, and can be connected to many clients at once. Usually kernels allow execution of only a single language, but there are a couple of exceptions. By default Jupyter Notebook ships with the Python kernel. As of the 2.3 release (October 2014), there are 49 Jupyter-compatible kernels for many programming languages, including Python, R, Julia and Haskell.

Code & Analysis:

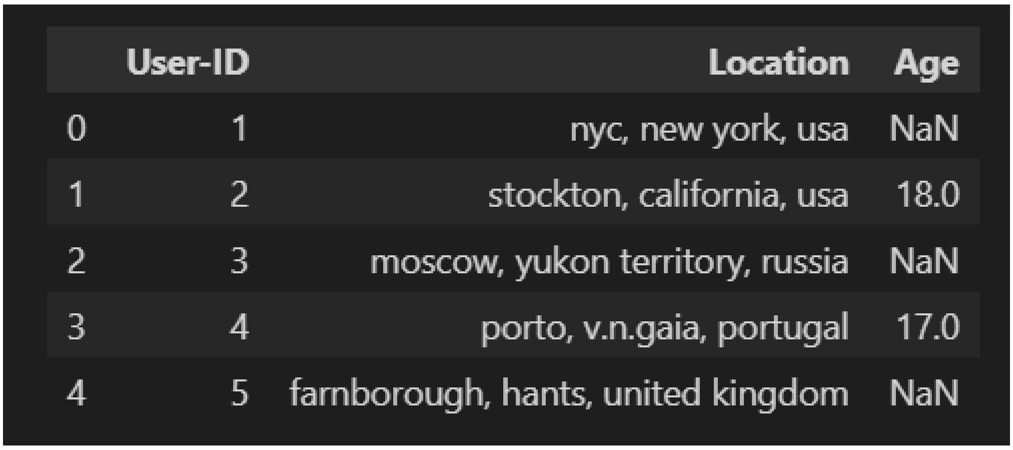
```
import numpy as np # linear algebra
import pandas as pd # data processing
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
import warnings; warnings.simplefilter('ignore')
```

```
book_data= pd.read_csv('BX-Books.csv', sep= ';', error_bad_lines= False,
encoding= 'latin-1')
book_data.head()
```



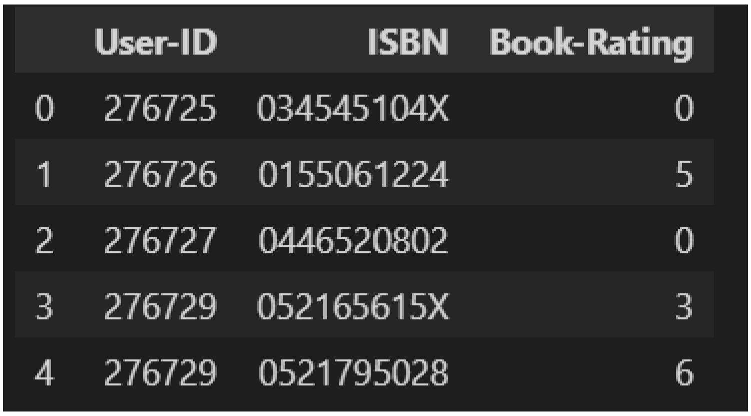
	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M	I
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	http://images.amazon.com/images/P/0195153448.0...	http://images.amazon.com/images/P/0195153448.0...	http://images.amazon.com/images/P/01
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.0...	http://images.amazon.com/images/P/0002005018.0...	http://images.amazon.com/images/P/00
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amazon.com/images/P/0060973129.0...	http://images.amazon.com/images/P/0060973129.0...	http://images.amazon.com/images/P/00
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux	http://images.amazon.com/images/P/0374157065.0...	http://images.amazon.com/images/P/0374157065.0...	http://images.amazon.com/images/P/03
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton &amp; Company	http://images.amazon.com/images/P/0393045218.0...	http://images.amazon.com/images/P/0393045218.0...	http://images.amazon.com/images/P/03

```
user_data= pd.read_csv('BX-Users.csv', sep= ';', error_bad_lines= False,
encoding= 'latin-1')
user_data.head()
```

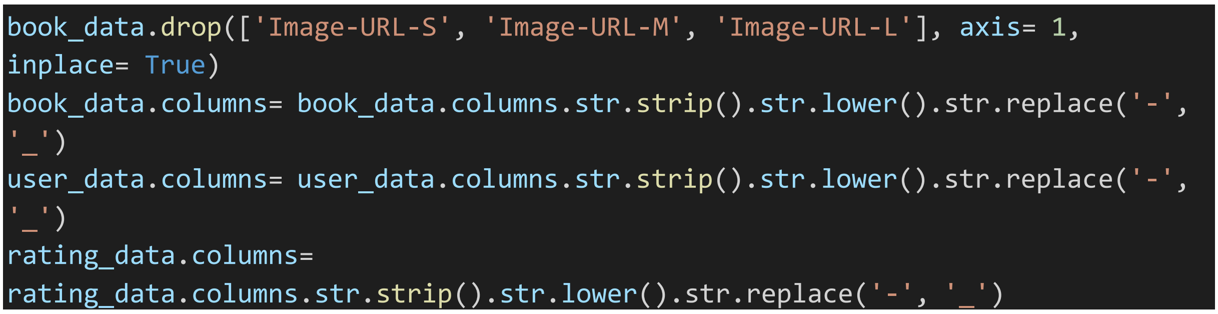


	User-ID	Location	Age
0	1	nyc, new york, usa	NaN
1	2	stockton, california, usa	18.0
2	3	moscow, yukon territory, russia	NaN
3	4	porto, v.n.gaia, portugal	17.0
4	5	farnborough, hants, united kingdom	NaN

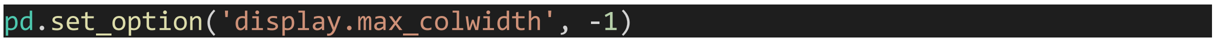
```
rating_data= pd.read_csv('BX-Book-Ratings.csv', sep= ';', error_bad_lines=
False, encoding= 'latin-1')
rating_data.head()
```



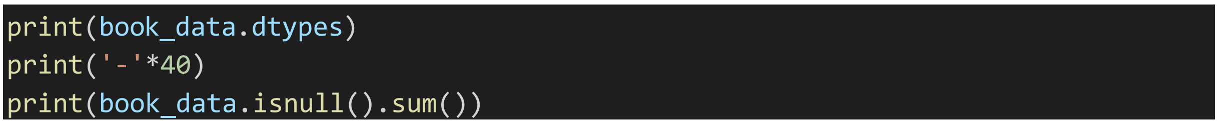
	User-ID	ISBN	Book-Rating
0	276725	034545104X	0
1	276726	0155061224	5
2	276727	0446520802	0
3	276729	052165615X	3
4	276729	0521795028	6



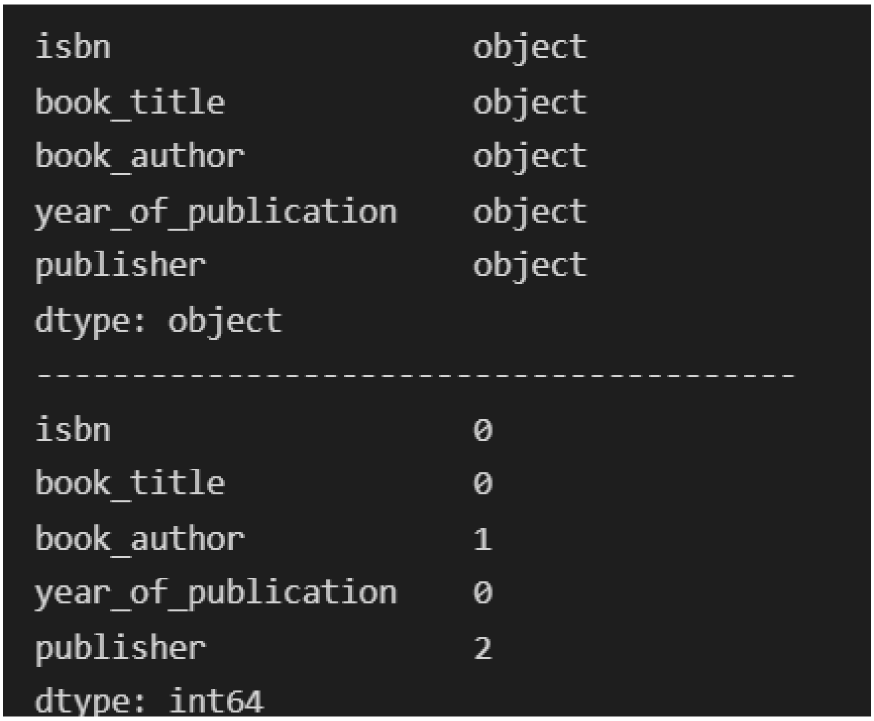
```
book_data.drop(['Image-URL-S', 'Image-URL-M', 'Image-URL-L'], axis= 1,
inplace= True)
book_data.columns= book_data.columns.str.strip().str.lower().str.replace('-',
'_')
user_data.columns= user_data.columns.str.strip().str.lower().str.replace('-',
'_')
rating_data.columns=
rating_data.columns.str.strip().str.lower().str.replace('-', '_')
```



```
pd.set_option('display.max_colwidth', -1)
```

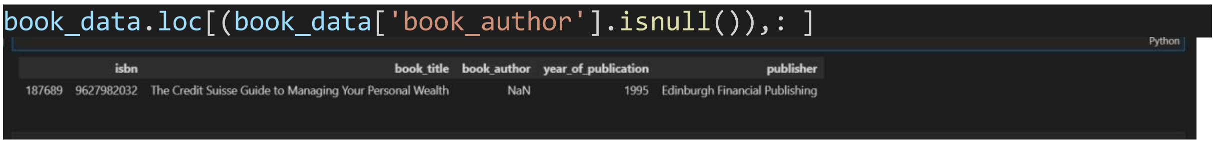


```
print(book_data.dtypes)
print('-'*40)
print(book_data.isnull().sum())
```



```
isbn                   object
book_title             object
book_author            object
year_of_publication    object
publisher              object
dtype: object
----------------------------------------
isbn                   0
book_title             0
book_author            1
year_of_publication    0
publisher              2
dtype: int64
```

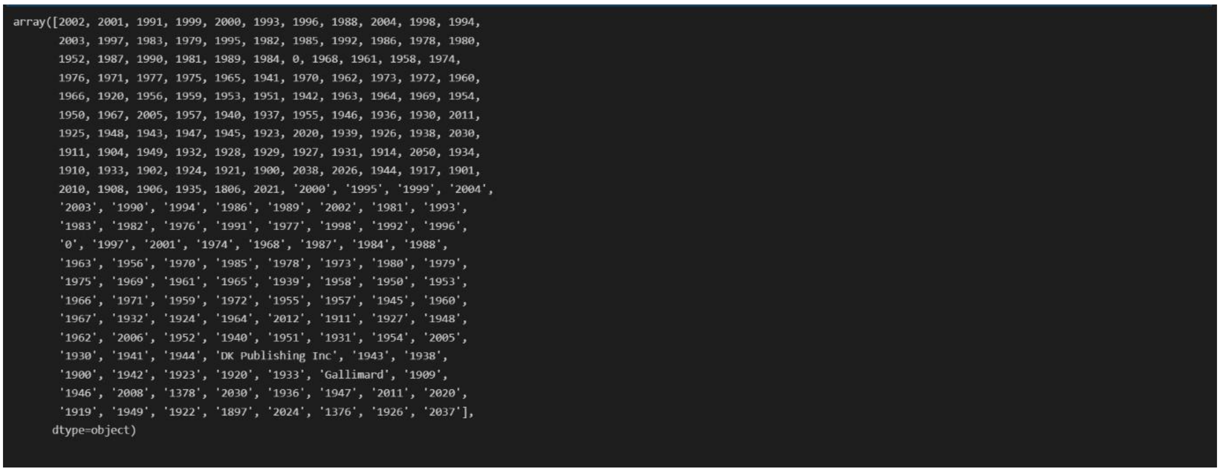
```python
book_data.loc[(book_data['book_author'].isnull()),: ]
```



	isbn	book_title	book_author	year_of_publication	publisher
187689	9627982032	The Credit Suisse Guide to Managing Your Personal Wealth	NaN	1995	Edinburgh Financial Publishing

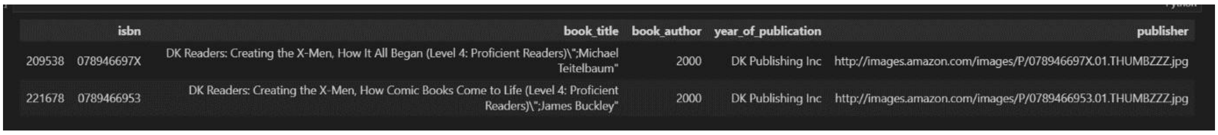
```python
book_data.loc[(book_data['isbn'] == '9627982032'),'book_author'] = 'Other'
```

```python
book_data['year_of_publication'].unique()
```



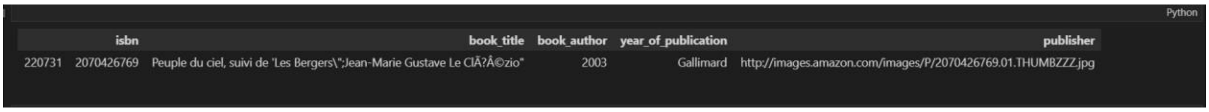
```
array([2002, 2001, 1991, 1999, 2000, 1993, 1996, 1988, 2004, 1998, 1994,
       2003, 1997, 1983, 1979, 1995, 1982, 1985, 1992, 1986, 1978, 1980,
       1952, 1987, 1990, 1981, 1989, 1984, 0, 1968, 1961, 1958, 1974,
       1976, 1971, 1977, 1975, 1965, 1941, 1970, 1962, 1973, 1972, 1960,
       1966, 1920, 1956, 1959, 1953, 1951, 1942, 1963, 1964, 1969, 1954,
       1950, 1967, 2005, 1957, 1940, 1937, 1955, 1946, 1936, 1930, 2011,
       1925, 1948, 1943, 1947, 1945, 1923, 2020, 1939, 1926, 1938, 2030,
       1911, 1904, 1949, 1932, 1928, 1929, 1927, 1931, 1914, 2050, 1934,
       1910, 1933, 1902, 1924, 1921, 1900, 2038, 2026, 1944, 1917, 1901,
       2010, 1908, 1906, 1935, 1806, 2021, '2000', '1995', '1999', '2004',
       '2003', '1990', '1994', '1986', '1989', '2002', '1981', '1993',
       '1983', '1982', '1976', '1991', '1977', '1998', '1992', '1996',
       '0', '1997', '2001', '1974', '1968', '1987', '1984', '1988',
       '1963', '1956', '1970', '1985', '1978', '1973', '1980', '1979',
       '1975', '1969', '1961', '1965', '1939', '1958', '1950', '1953',
       '1966', '1971', '1959', '1972', '1955', '1957', '1945', '1960',
       '1967', '1932', '1924', '1964', '2012', '1911', '1927', '1948',
       '1962', '2006', '1952', '1940', '1951', '1931', '1954', '2005',
       '1930', '1941', '1944', 'DK Publishing Inc', '1943', '1938',
       '1900', '1942', '1923', '1920', '1933', 'Gallimard', '1909',
       '1946', '2008', '1378', '2030', '1936', '1947', '2011', '2020',
       '1919', '1949', '1922', '1897', '2024', '1376', '1926', '2037'],
      dtype=object)
```

```python
book_data[book_data['year_of_publication'] == 'DK Publishing Inc']
```



	isbn	book_title	book_author	year_of_publication	publisher
209538	078946697X	DK Readers: Creating the X-Men, How It All Began (Level 4: Proficient Readers)\";Michael Teitelbaum"	2000	DK Publishing Inc	http://images.amazon.com/images/P/078946697X.01.THUMBZZZ.jpg
221678	0789466953	DK Readers: Creating the X-Men, How Comic Books Come to Life (Level 4: Proficient Readers)\";James Buckley"	2000	DK Publishing Inc	http://images.amazon.com/images/P/0789466953.01.THUMBZZZ.jpg

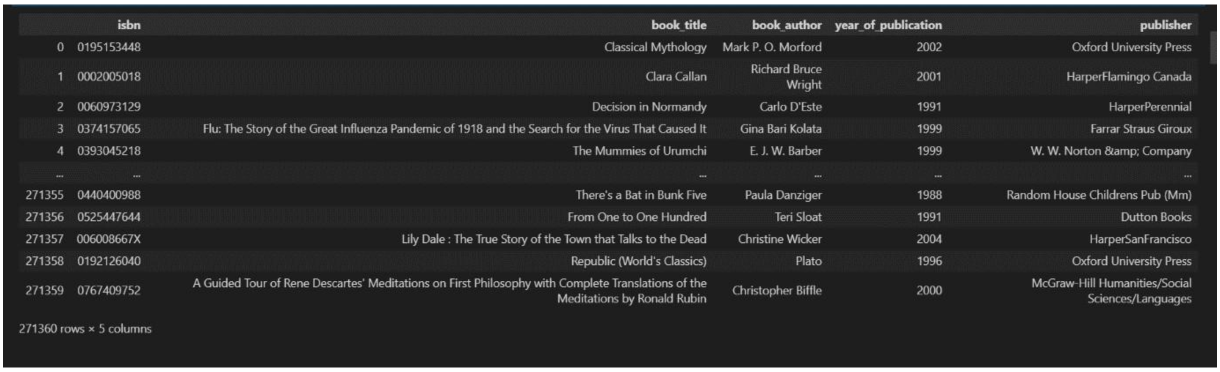
```python
book_data[book_data['year_of_publication'] == 'Gallimard']
```



	isbn	book_title	book_author	year_of_publication	publisher
220731	2070426769	Peuple du ciel, suivi de 'Les Bergers\";Jean-Marie Gustave Le ClÃ?Â©zio"	2003	Gallimard	http://images.amazon.com/images/P/2070426769.01.THUMBZZZ.jpg

```python
def replace_df_value(df, idx, col_name, val):
    df.loc[idx, col_name] = val
    return df
```

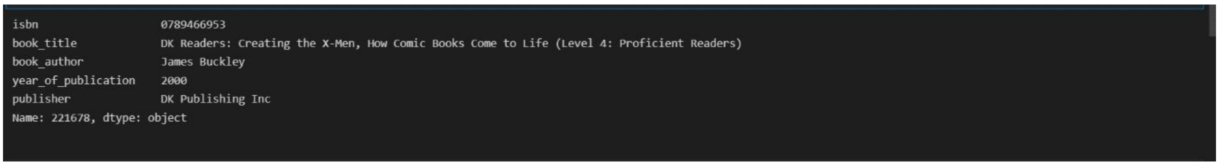
```python
replace_df_value(book_data, 209538, 'book_title', 'DK Readers: Creating the X-Men, How It All Began (Level 4: Proficient Readers)')
replace_df_value(book_data, 209538, 'book_author', 'Michael Teitelbaum')
replace_df_value(book_data, 209538, 'year_of_publication', 2000)
replace_df_value(book_data, 209538, 'publisher', 'DK Publishing Inc')

replace_df_value(book_data, 221678, 'book_title', 'DK Readers: Creating the X-Men, How Comic Books Come to Life (Level 4: Proficient Readers)')
replace_df_value(book_data, 221678, 'book_author', 'James Buckley')
replace_df_value(book_data, 221678, 'year_of_publication', 2000)
replace_df_value(book_data, 221678, 'publisher', 'DK Publishing Inc')

replace_df_value(book_data, 220731,'book_title', "Peuple du ciel, suivi de 'Les Bergers")
replace_df_value(book_data, 220731, 'book_author', 'Jean-Marie Gustave Le ClÃ?Â©zio')
replace_df_value(book_data, 220731, 'year_of_publication', 2003)
replace_df_value(book_data, 220731, 'publisher', 'Gallimard')
```



	isbn	book_title	book_author	year_of_publication	publisher
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial
3	0374157065	Flu: The Story of the Great Influenza Pandemic of 1918 and the Search for the Virus That Caused It	Gina Bari Kolata	1999	Farrar Straus Giroux
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton &amp; Company
...	...	...	...	...	...
271355	0440400988	There's a Bat in Bunk Five	Paula Danziger	1988	Random House Childrens Pub (Mm)
271356	0525447644	From One to One Hundred	Teri Sloat	1991	Dutton Books
271357	006008667X	Lily Dale : The True Story of the Town that Talks to the Dead	Christine Wicker	2004	HarperSanFrancisco
271358	0192126040	Republic (World's Classics)	Plato	1996	Oxford University Press
271359	0767409752	A Guided Tour of Rene Descartes' Meditations on First Philosophy with Complete Translations of the Meditations by Ronald Rubin	Christopher Biffle	2000	McGraw-Hill Humanities/Social Sciences/Languages

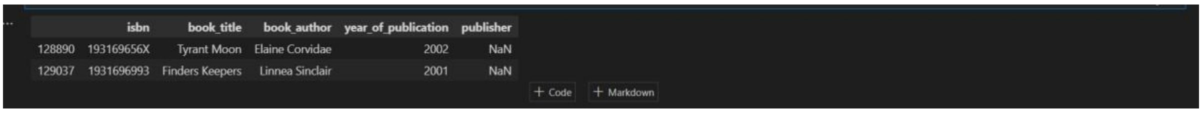
271360 rows × 5 columns

```python
book_data.loc[221678]
```



```
isbn                   0789466953
book_title             DK Readers: Creating the X-Men, How Comic Books Come to Life (Level 4: Proficient Readers)
book_author            James Buckley
year_of_publication    2000
publisher              DK Publishing Inc
Name: 221678, dtype: object
```

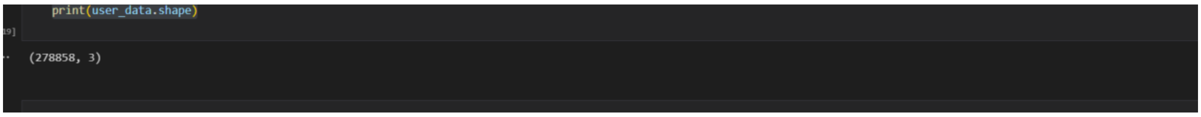
```python
book_data.loc[(book_data['publisher'].isnull()),: ]
```



	isbn	book_title	book_author	year_of_publication	publisher
128890	193169656X	Tyrant Moon	Elaine Corvidae	2002	NaN
129037	1931696993	Finders Keepers	Linnea Sinclair	2001	NaN

```python
book_data.loc[(book_data['isbn'] == '193169656X'),'publisher'] = 'No Mention'
book_data.loc[(book_data['isbn'] == '1931696993'),'publisher'] = 'No Mention'
```

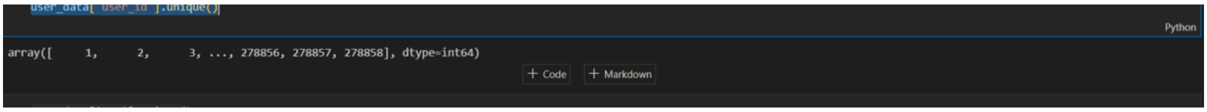
```
print(user_data.shape)
```



```
print(user_data.shape)
```

```
(278858, 3)
```

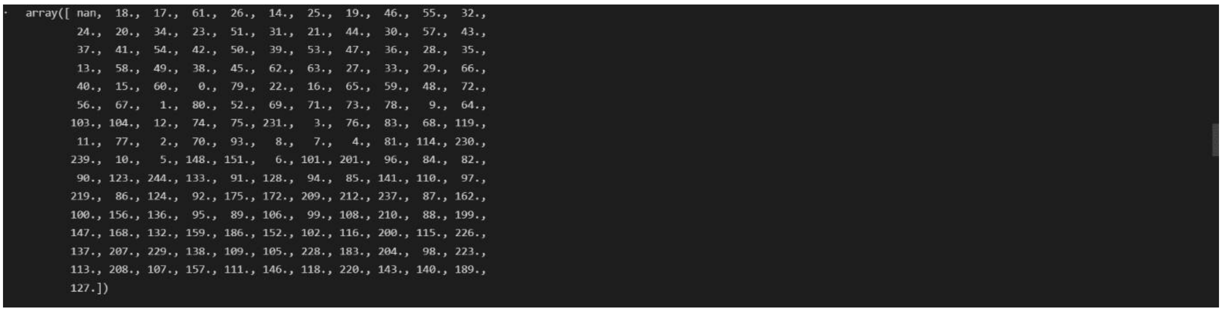
```
user_data['user_id'].unique()
```



```
user_data['user_id'].unique()
```

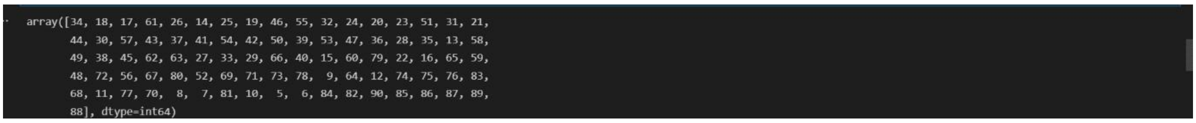
```
array([     1,      2,      3, ..., 278856, 278857, 278858], dtype=int64)
```

```
user_data['age'].unique()
```



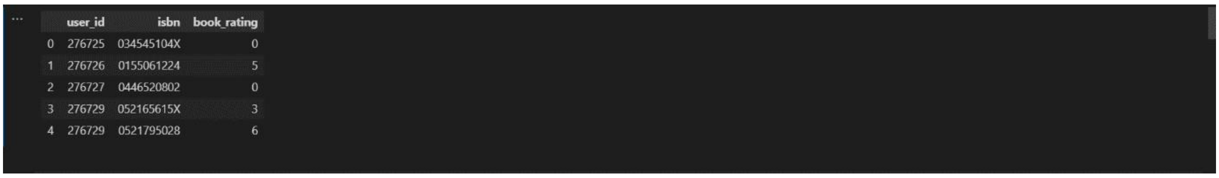
```
array([ nan,  18.,  17.,  61.,  26.,  14.,  25.,  19.,  46.,  55.,  32.,
        24.,  20.,  34.,  23.,  51.,  31.,  21.,  44.,  30.,  57.,  43.,
        37.,  41.,  54.,  42.,  50.,  39.,  53.,  47.,  36.,  28.,  35.,
        13.,  58.,  49.,  38.,  45.,  62.,  63.,  27.,  33.,  29.,  66.,
        40.,  15.,  60.,   0.,  79.,  22.,  16.,  65.,  59.,  48.,  72.,
        56.,  67.,   1.,  80.,  52.,  69.,  71.,  73.,  78.,   9.,  64.,
       103., 104.,  12.,  74.,  75., 231.,   3.,  76.,  83.,  68., 119.,
        11.,  77.,   2.,  70.,  93.,   8.,   7.,   4.,  81., 114., 230.,
       239.,  10.,   5., 148., 151.,   6., 101., 201.,  96.,  84.,  82.,
        90., 123., 244., 133.,  91., 128.,  94.,  85., 141., 110.,  97.,
       219.,  86., 124.,  92., 175., 172., 209., 212., 237.,  87., 162.,
       100., 156., 136.,  95.,  89., 106.,  99., 108., 210.,  88., 199.,
       147., 168., 132., 159., 186., 152., 102., 116., 200., 115., 226.,
       137., 207., 229., 138., 109., 105., 228., 183., 204.,  98., 223.,
       113., 208., 107., 157., 111., 146., 118., 220., 143., 140., 189.,
       127.])
```

```
user_data.loc[(user_data['age'] > 90) | (user_data['age'] < 5)] = np.nan
user_data['age'].fillna((user_data['age'].mean()), inplace=True)
user_data['age']= user_data['age'].astype('int64')
user_data['age'].unique()
```



```
array([34, 18, 17, 61, 26, 14, 25, 19, 46, 55, 32, 24, 20, 23, 51, 31, 21,
       44, 30, 57, 43, 37, 41, 54, 42, 50, 39, 53, 47, 36, 28, 35, 13, 58,
       49, 38, 45, 62, 63, 27, 33, 29, 66, 40, 15, 60, 79, 22, 16, 65, 59,
       48, 72, 56, 67, 80, 52, 69, 71, 73, 78,  9, 64, 12, 74, 75, 76, 83,
       68, 11, 77, 70,  8,  7, 81, 10,  5,  6, 84, 82, 90, 85, 86, 87, 89,
       88], dtype=int64)
```

```
rating_data.head()
```

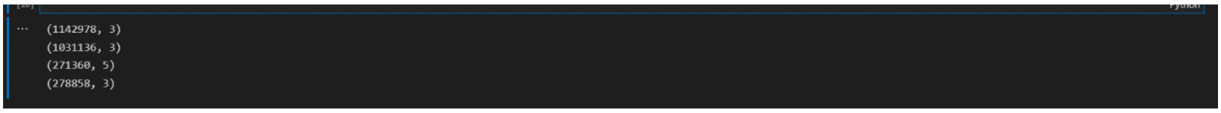


	user_id	isbn	book_rating
0	276725	034545104X	0
1	276726	0155061224	5
2	276727	0446520802	0
3	276729	052165615X	3
4	276729	0521795028	6

```
unique_ratings = rating_data[rating_data.isbn.isin(book_data.isbn)]
```

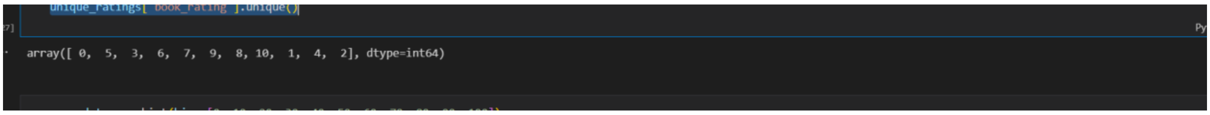
```
rating_data = rating_data[rating_data.user_id.isin(user_data.user_id)]
```

```
print(rating_data.shape)
print(unique_ratings.shape)
print(book_data.shape)
print(user_data.shape)
```



```
(1142978, 3)
(1031136, 3)
(271360, 5)
(278858, 3)
```

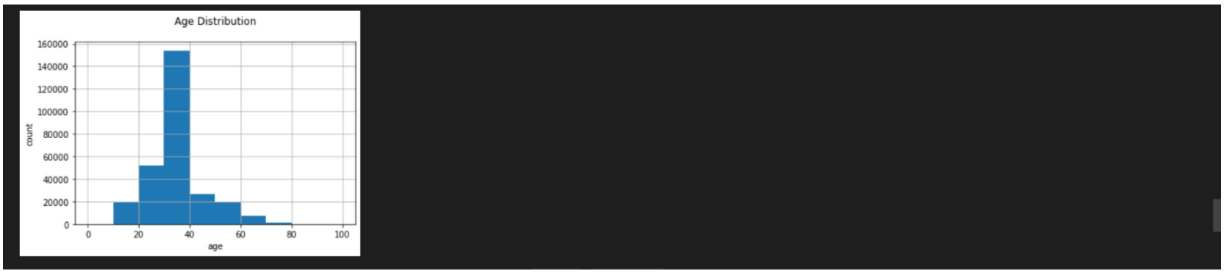
```
unique_ratings['book_rating'].unique()
```



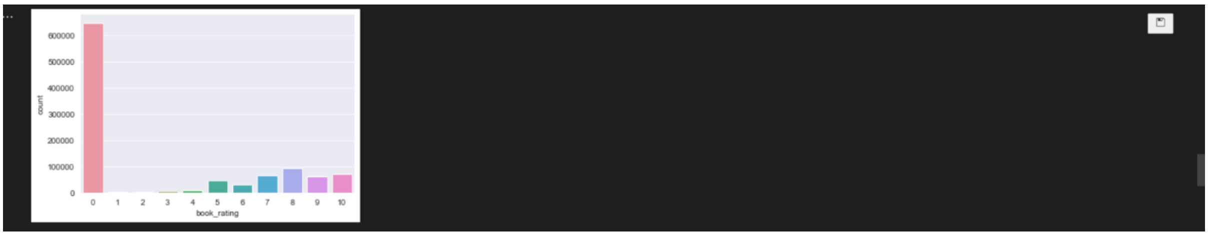
```
unique_ratings['book_rating'].unique()
```

```
array([ 0,  5,  3,  6,  7,  9,  8, 10,  1,  4,  2], dtype=int64)
```

```
user_data.age.hist(bins=[0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100])
plt.title('Age Distribution\n')
plt.xlabel('age')
plt.ylabel('count')
plt.savefig('age_dist.png', bbox_inches='tight')
plt.show()
```

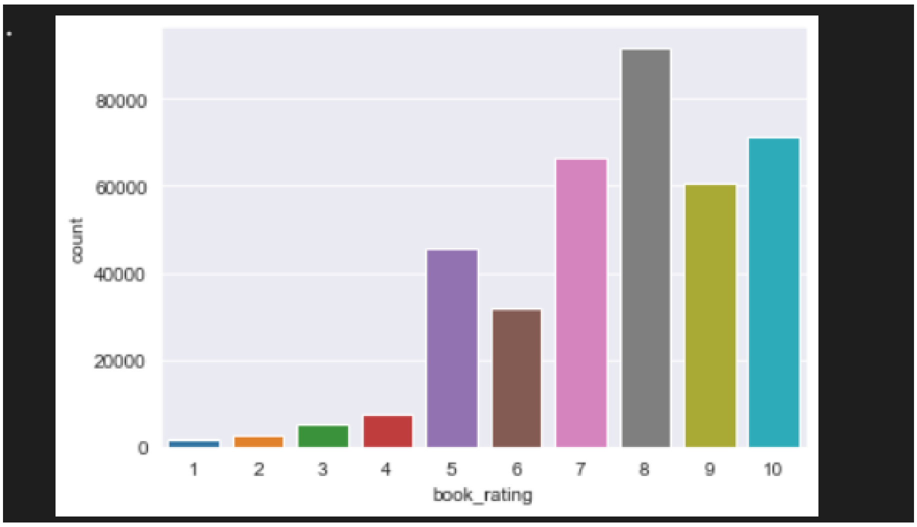


```
sns.set_style('darkgrid')
sns.countplot(data= unique_ratings , x='book_rating')
plt.show()
```

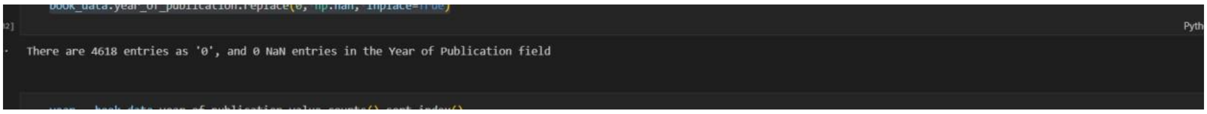


```
ratings_explicit= unique_ratings[unique_ratings['book_rating'] != 0]
ratings_implicit= unique_ratings[unique_ratings['book_rating'] == 0]
```

```
sns.set_style('darkgrid')
sns.countplot(data= ratings_explicit , x='book_rating')
plt.show()
```

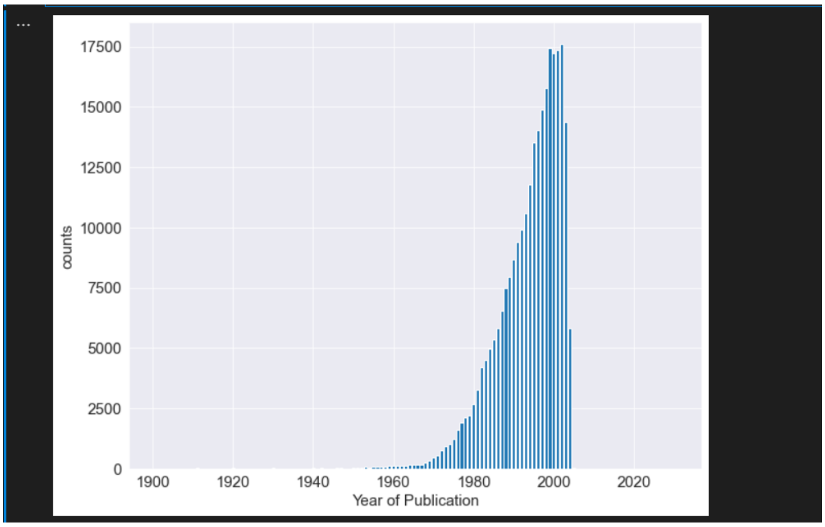


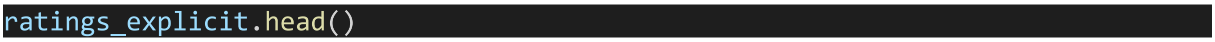
```python
book_data.year_of_publication = pd.to_numeric(book_data.year_of_publication,
errors='coerce')

# Check for 0's or NaNs in Year of Publication
zero_year = book_data[book_data.year_of_publication ==
0].year_of_publication.count()
nan_year = book_data.year_of_publication.isnull().sum()

print(f'There are {zero_year} entries as \'0\', and {nan_year} NaN entries in
the Year of Publication field')

# Replace all years of zero with NaN
book_data.year_of_publication.replace(0, np.nan, inplace=True)
```



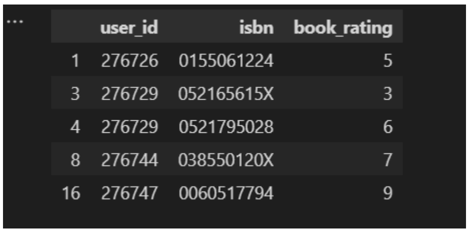
```
There are 4618 entries as '0', and 0 NaN entries in the Year of Publication field
```

```python
year = book_data.year_of_publication.value_counts().sort_index()
year = year.where(year>5)
plt.figure(figsize=(10, 8))
plt.rcParams.update({'font.size': 15})
plt.bar(year.index, year.values)
plt.xlabel('Year of Publication')
plt.ylabel('counts')
plt.show()
```

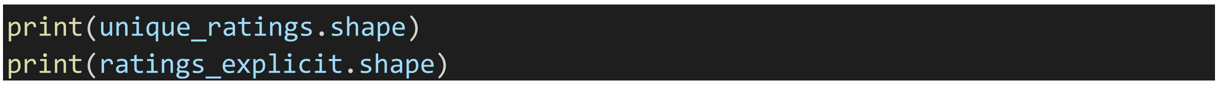




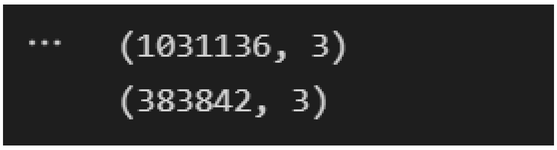
```
ratings_explicit.head()
```



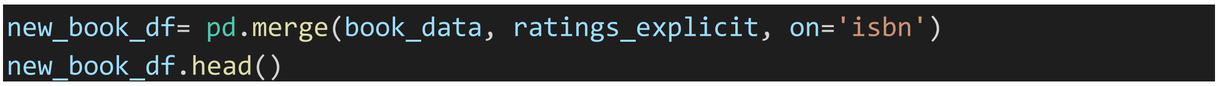
	user_id	isbn	book_rating
1	276726	0155061224	5
3	276729	052165615X	3
4	276729	0521795028	6
8	276744	038550120X	7
16	276747	0060517794	9



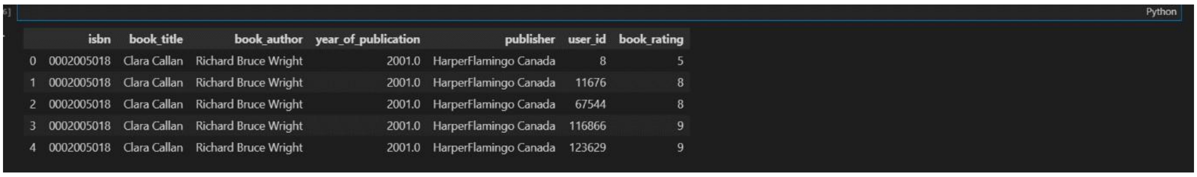
```
print(unique_ratings.shape)
print(ratings_explicit.shape)
```



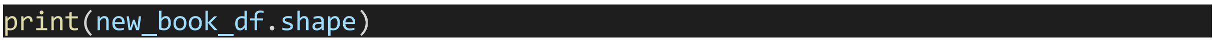
```
(1031136, 3)
(383842, 3)
```



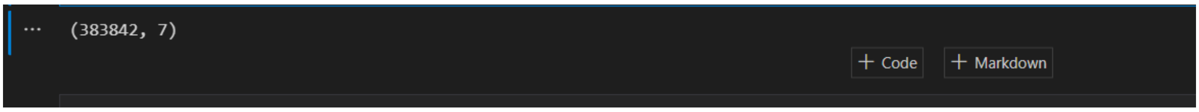
```
new_book_df= pd.merge(book_data, ratings_explicit, on='isbn')
new_book_df.head()
```



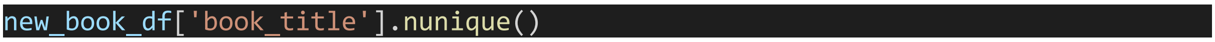
	isbn	book_title	book_author	year_of_publication	publisher	user_id	book_rating
0	0002005018	Clara Callan	Richard Bruce Wright	2001.0	HarperFlamingo Canada	8	5
1	0002005018	Clara Callan	Richard Bruce Wright	2001.0	HarperFlamingo Canada	11676	8
2	0002005018	Clara Callan	Richard Bruce Wright	2001.0	HarperFlamingo Canada	67544	8
3	0002005018	Clara Callan	Richard Bruce Wright	2001.0	HarperFlamingo Canada	116866	9
4	0002005018	Clara Callan	Richard Bruce Wright	2001.0	HarperFlamingo Canada	123629	9



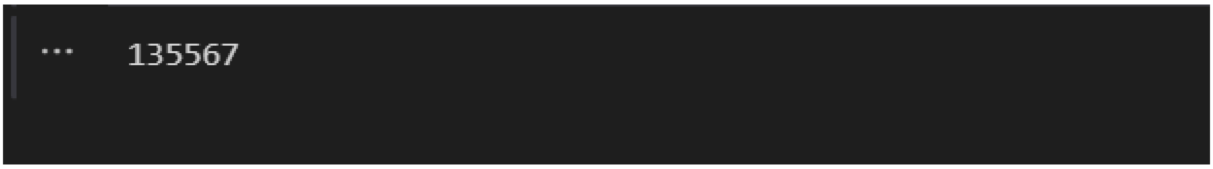
```
print(new_book_df.shape)
```



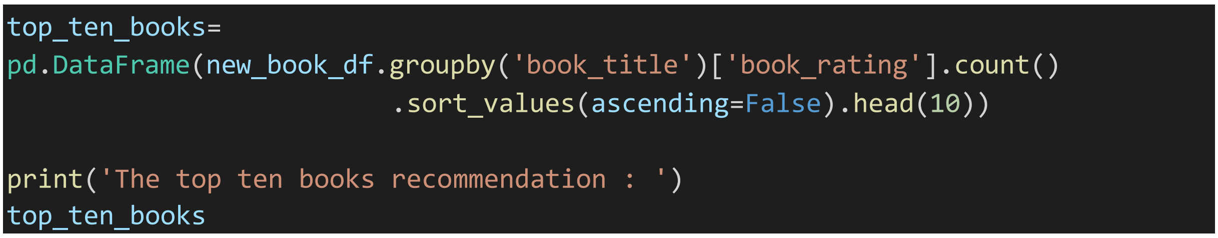
```
(383842, 7)
```



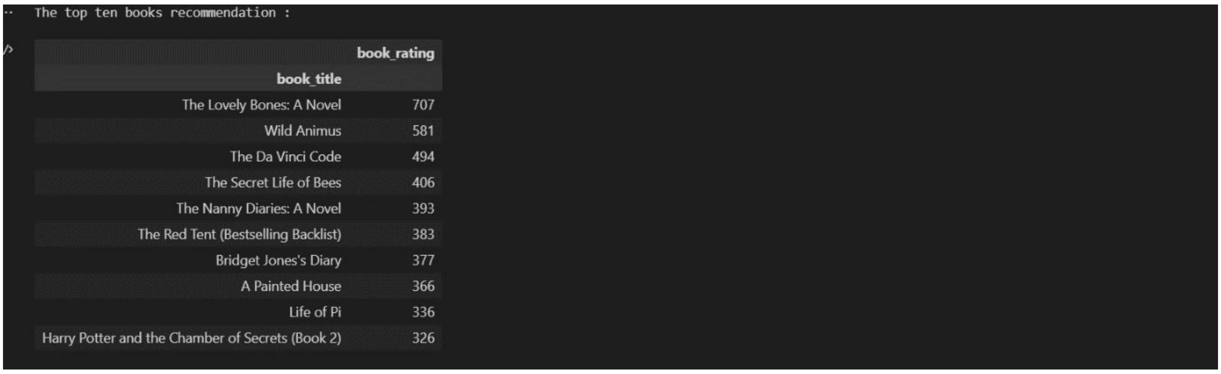
```
new_book_df['book_title'].nunique()
```



```
...   135567
```

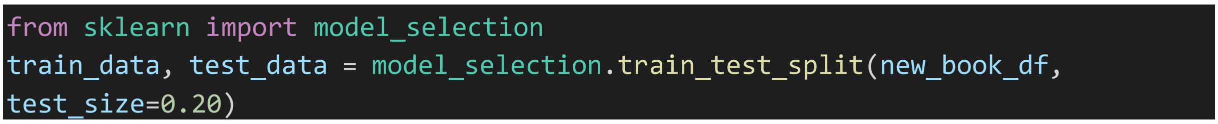


```
top_ten_books=
pd.DataFrame(new_book_df.groupby('book_title')['book_rating'].count()
                          .sort_values(ascending=False).head(10))

print('The top ten books recommendation : ')
top_ten_books
```

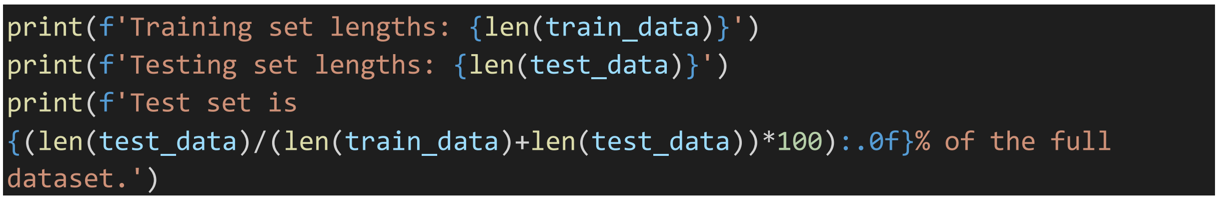


```
The top ten books recommendation :
```

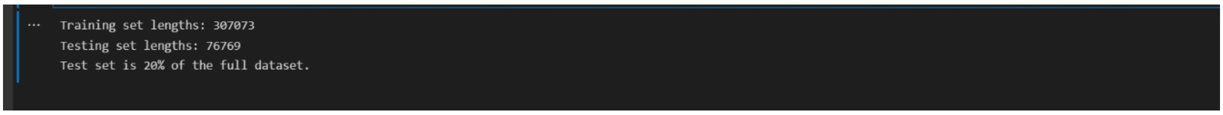
	book_rating
book_title	
The Lovely Bones: A Novel	707
Wild Animus	581
The Da Vinci Code	494
The Secret Life of Bees	406
The Nanny Diaries: A Novel	393
The Red Tent (Bestselling Backlist)	383
Bridget Jones's Diary	377
A Painted House	366
Life of Pi	336
Harry Potter and the Chamber of Secrets (Book 2)	326



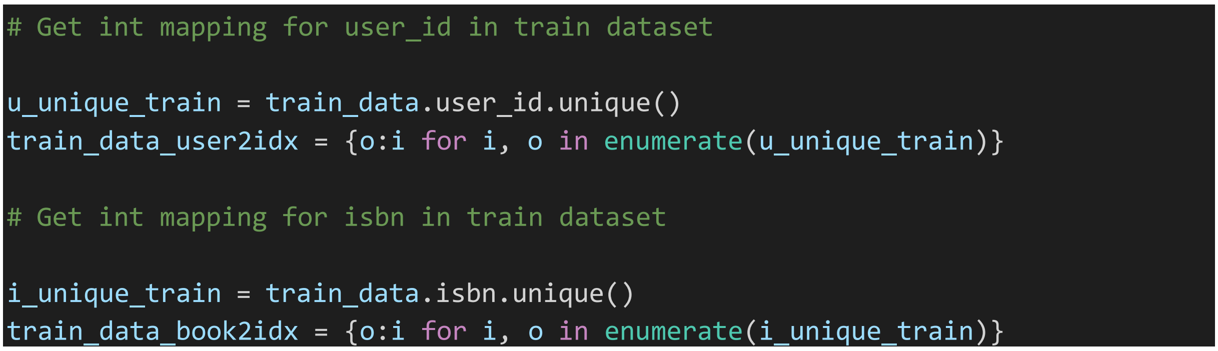
```
from sklearn import model_selection
train_data, test_data = model_selection.train_test_split(new_book_df,
test_size=0.20)
```



```
print(f'Training set lengths: {len(train_data)}')
print(f'Testing set lengths: {len(test_data)}')
print(f'Test set is
{(len(test_data)/(len(train_data)+len(test_data))*100):.0f}% of the full
dataset.')
```



```
...  Training set lengths: 307073
     Testing set lengths: 76769
     Test set is 20% of the full dataset.
```

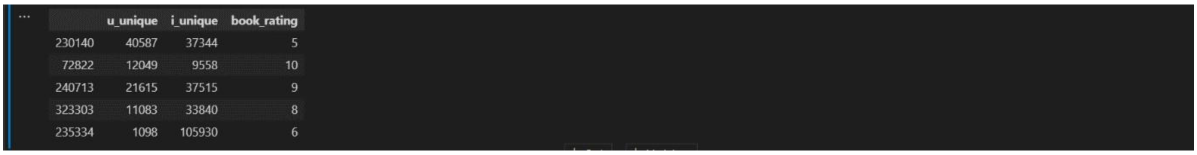


```
# Get int mapping for user_id in train dataset

u_unique_train = train_data.user_id.unique()
train_data_user2idx = {o:i for i, o in enumerate(u_unique_train)}

# Get int mapping for isbn in train dataset

i_unique_train = train_data.isbn.unique()
train_data_book2idx = {o:i for i, o in enumerate(i_unique_train)}
```

```python
# Get int mapping for user_id in test dataset

u_unique_test = test_data.user_id.unique()
test_data_user2idx = {o:i for i, o in enumerate(u_unique_test)}

# Get int mapping for isbn in train dataset

i_unique_test = test_data.isbn.unique()
test_data_book2idx = {o:i for i, o in enumerate(i_unique_test)}
```

```python
# TRAINING SET
train_data['u_unique'] = train_data['user_id'].map(train_data_user2idx)
train_data['i_unique'] = train_data['isbn'].map(train_data_book2idx)

# TESTING SET
test_data['u_unique'] = test_data['user_id'].map(test_data_user2idx)
test_data['i_unique'] = test_data['isbn'].map(test_data_book2idx)

# Convert back to 3-column df
train_data = train_data[['u_unique', 'i_unique', 'book_rating']]
test_data = test_data[['u_unique', 'i_unique', 'book_rating']]
```

```python
train_data.sample(5)
```



	u_unique	i_unique	book_rating
230140	40587	37344	5
72822	12049	9558	10
240713	21615	37515	9
323303	11083	33840	8
235334	1098	105930	6

Project Outcome: -

From this project, we learnt to describe a flow process for data science problems and classified data science problems into standard typology. We also learnt about correlating results to the solution approach followed and assessing the solution approach.

Project Conclusion: -

From this project, we gained the knowledge of software – Jupyter Notebook & Python Language. We learnt to the algorithm behind Book Recommendation system & learned abiut Matrix Factorization, and also did analysis of the whole process.