In [ ]:
```
import gdown

url = 'https://drive.google.com/uc?id=1PL13wgXLfXcsrkKNuVIaNJdGXrIqv2mv'
output = 'book_crossing.cleaned.csv'
gdown.download(url, output, quiet=False)
```

```
Downloading...
From: https://drive.google.com/uc?id=1PL13wgXLfXcsrkKNuVIaNJdGXrIqv2mv
To: /content/book_crossing.cleaned.csv
44.9MB [00:01, 40.7MB/s]
```

Out[ ]:
```
'book_crossing.cleaned.csv'
```

In [ ]:
```
%matplotlib inline

import scipy
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

sns.set()
palette = sns.color_palette("icefire")

plt.style.use('ggplot')

sns.set_context("talk")
```

BookCrossing - Cleaning

In [ ]:
```
dataset = pd.read_csv('book_crossing.cleaned.csv')
```

In [ ]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 383849 entries, 0 to 383848
Data columns (total 11 columns):
 #   Column               Non-Null Count   Dtype 
---  ------               --------------   ----- 
 0   user_id              383849 non-null  int64 
 1   age                  383849 non-null  int64 
 2   isbn                 383849 non-null  object
 3   book_rating          383849 non-null  int64 
 4   book_title           383849 non-null  object
 5   book_author          383849 non-null  object
 6   year_of_publication  383849 non-null  int64 
 7   publisher            383849 non-null  object
 8   city                 375223 non-null  object
 9   state                371248 non-null  object
 10  country              366406 non-null  object
dtypes: int64(4), object(7)
memory usage: 32.2+ MB
```

We won't be considering city, state, because they don't really tell a lot of the rating of a book, but also most of the users (~70%) are from usa (which may not contribute a lot to accuracy of classification, but we'll consider it), and the location is realated to the user, and not the book directly, we'll also be dropping isbn, user_id, since they dont contribute to classification of rating

In [ ]:
```
dataset = dataset.drop(['user_id', 'isbn', 'city', 'state'], axis=1)
```

In [ ]:
```
f'Dataset Shape : {dataset.shape}'
```

Out[ ]:
```
'Dataset Shape : (383849, 7)'
```

In [ ]:
```
dataset.dropna(inplace=True)
```

In [ ]:
```
f'Dataset Shape after dropping NA: {dataset.shape}'
```

Out[ ]:
```
'Dataset Shape after dropping NA: (366406, 7)'
```

In [ ]:
```
dataset.head()
```

Out[ ]:

	age	book_rating	book_title	book_author	year_of_publication	publisher	country
0	34	5	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
2	30	8	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
4	34	9	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
5	34	8	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
6	34	9	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada

In [ ]:
```
dataset.describe().T
```

Out[ ]:

	count	mean	std	min	25%	50%	75%	max
age	366406.0	35.860998	10.448608	5.0	30.0	34.0	40.0	100.0
book_rating	366406.0	7.635975	1.836354	1.0	7.0	8.0	9.0	10.0
year_of_publication	366406.0	1995.670314	7.397156	1376.0	1993.0	1997.0	2001.0	2006.0

We'll remove the rows which have a country which has value count <= 50

In [ ]:
```
dataset = dataset.groupby('country').filter(lambda x: len(x) > 50)
```

In [ ]:
```
dataset.describe().T
```

Out[ ]:

	count	mean	std	min	25%	50%	75%	max
age	364570.0	35.867227	10.447887	5.0	30.0	34.0	40.0	100.0
book_rating	364570.0	7.636709	1.835857	1.0	7.0	8.0	9.0	10.0
year_of_publication	364570.0	1995.667164	7.400552	1376.0	1993.0	1997.0	2001.0	2006.0

In [ ]:
```
f'Dataset Shape : {dataset.shape}'
```

Out[ ]:
```
'Dataset Shape : (364570, 7)'
```

In [ ]:
```
f'Column Names: {dataset.columns.to_list()}'
```

Out[ ]:
```
"Column Names: ['age', 'book_rating', 'book_title', 'book_author', 'year_of_publication', 'publisher', 'country']"
```

In [ ]:
```
dataset['book_rating'].value_counts()
```

Out[ ]:
```
8     87090
10    68038
7     63036
9     58080
5     42988
6     29943
4      7120
3      4746
2      2198
1      1331
Name: book_rating, dtype: int64
```

We'll now convert the rating into classification categories

In [ ]:
```
bins = [0, 3, 7, 10]
names = ['low', 'mid', 'high']

dataset['book_rating'] = pd.cut(dataset['book_rating'], bins, labels=names)
```

In [ ]:
```
dataset.head()
```

Out[ ]:

	age	book_rating	book_title	book_author	year_of_publication	publisher	country
0	34	mid	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
2	30	high	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
4	34	high	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
5	34	high	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
6	34	high	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada

In [ ]:
```
dataset['book_rating'].value_counts()
```

Out[ ]:
```
high    213208
mid     143087
low       8275
Name: book_rating, dtype: int64
```

In [ ]:
```
dataset.to_csv('book_crossing.classification.cleaned.csv', index=False)
```