```
## Loading the dataset
import pandas as pd
data = pd.read_csv('/content/all_kindle_review.csv')
```

```
data
```

	Unnamed: 0.1	Unnamed: 0	asin	helpful	rating	reviewText	reviewTime	reviewerID	reviewerName	summary
0	0	11539	B0033UV8HI	[8, 10]	3	Jace Rankin may be short, but he's nothing to ...	09 2, 2010	A3HHXRELK8BHQG	Ridley	Entertaining But Average
1	1	5957	B002HJV4DE	[1, 1]	5	Great short read. I didn't want to put it dow...	10 8, 2013	A2RGNZ0TRF578I	Holly Butler	Terrific menage scenes!
2	2	9146	B002ZG96I4	[0, 0]	3	I'll start by saying this is the first of four...	04 11, 2014	A3S0H2HV6U1I7F	Merissa	Snapdragon Alley
3	3	7038	B002QHWOEU	[1, 3]	3	Aggie is Angela Lansbury who carries pocketboo...	07 5, 2014	AC4OQW3GZ919J	Cleargrace	very light murder cozy
4	4	1776	B001A06VJ8	[0, 1]	4	I did not expect this type of book to be in li...	12 31, 2012	A3C9V987IQHOQD	Rjostler	Book
...	...	...	...	...	...	...	...	...	...	...
11995	11995	2183	B001DUGORO	[0, 0]	4	Valentine cupid is a vampire-Jena and Ian ano...	02 28, 2014	A1OKS5Q1HD8WQC	lisa jon jung	jena
11996	11996	6272	B002JCSFSQ	[2, 2]	5	I have read all seven books in this series. Ap...	05 16, 2011	AQRSPXLNEQAMA	TerryLP	Peacekeepers Series
11997	11997	12483	B0035N1V7K	[0, 1]	3	This book really just wasn't my cuppa. The si...	07 26, 2013	A2T5QLT5VXOJAK	hwilson	a little creepy
11998	11998	3640	B001W1XT40	[1, 2]	1	tried to use it to charge my kindle, it didn't...	09 17, 2013	A28MHD2DDY6DXB	Allison A. Slater "Gryphon50"	didn't work
11999	11999	11398	B003370JUS	[5, 6]	3	Taking Instruction is a look into the often hi...	07 5, 2012	A3JUXLB4K9ZXCC	Dafna Yee	If you like BDSM with a touch of romance, this...

12000 rows × 11 columns

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
df = data[['reviewText','rating']]
```

```
df.head()
```

	reviewText	rating
0	Jace Rankin may be short, but he's nothing to ...	3
1	Great short read. I didn't want to put it dow...	5
2	I'll start by saying this is the first of four...	3
3	Aggie is Angela Lansbury who carries pocketboo...	3
4	I did not expect this type of book to be in li...	4

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.shape
```

```
(12000, 2)
```

```
## Missing values
df.isnull().sum()
```

	0
reviewText	0
rating	0

```
## Unique values
df['rating'].unique()
```

```
array([3, 5, 4, 2, 1])
```

```
df['rating'].value_counts()
```

rating	count
5	3000
4	3000
3	2000
2	2000
1	2000

```
## Preprocesssing and Cleaning
## Positive review is 1 and negative review is 0
df.loc[:, 'rating'] = df['rating'].apply(lambda x : 0 if x<3 else 1 )
```

```
df['rating'].value_counts()
```

rating	count
1	8000
0	4000

```
## Preprocessing Steps
## 1. Lower all text
```

```
df['reviewText'] = df['reviewText'].str.lower()
df.head()
```

```
<ipython-input-14-cc7f31db30b9>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
  df['reviewText'] = df['reviewText'].str.lower()
```

	reviewText	rating
0	jace rankin may be short, but he's nothing to ...	1
1	great short read. i didn't want to put it dow...	1
2	i'll start by saying this is the first of four...	1
3	aggie is angela lansbury who carries pocketboo...	1
4	i did not expect this type of book to be in li...	1

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
## Removing Special Characters
import re
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
from bs4 import BeautifulSoup
```

```
## Removing special characters
df['reviewText']=df['reviewText'].apply(lambda x:re.sub('[^a-z A-z 0-9-]+', '',x))
## Remove the stopswords
df['reviewText']=df['reviewText'].apply(lambda x:" ".join([y for y in x.split() if y not in stopwords.words('english')]))
## Remove url
df['reviewText']=df['reviewText'].apply(lambda x: re.sub(r'(http|https|ftp|ssh):///([\w_-]+(?:(?:\.[\w_-]+)+))([\w.,@?^=%&:/~+#-
## Remove html tags
df['reviewText']=df['reviewText'].apply(lambda x: BeautifulSoup(x, 'lxml').get_text())
## Remove any additional spaces
df['reviewText']=df['reviewText'].apply(lambda x: " ".join(x.split()))
```

```
<ipython-input-17-86b4aeb70b16>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
  df['reviewText']=df['reviewText'].apply(lambda x:re.sub('[^a-z A-z 0-9-]+', '',x))
<ipython-input-17-86b4aeb70b16>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
  df['reviewText']=df['reviewText'].apply(lambda x:" ".join([y for y in x.split() if y not in stopwords.words('english')]))
<ipython-input-17-86b4aeb70b16>:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
  df['reviewText']=df['reviewText'].apply(lambda x: re.sub(r'(http|https|ftp|ssh):///([\w_-]+(?:(?:\.[\w_-]+)+))([\w.,@?^=%&:/~
<ipython-input-17-86b4aeb70b16>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
  df['reviewText']=df['reviewText'].apply(lambda x: BeautifulSoup(x, 'lxml').get_text())
<ipython-input-17-86b4aeb70b16>:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
  df['reviewText']=df['reviewText'].apply(lambda x: " ".join(x.split()))
```

```
df.head()
```

	reviewText	rating
0	jace rankin may short hes nothing mess man hau...	1
1	great short read didnt want put read one sitti...	1
2	ill start saying first four books wasnt expect...	1
3	aggie angela lansbury carries pocketbooks inst...	1
4	expect type book library pleased find price right	1

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
## Lematizer
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
True
```

```
def lemmatize_words(text):
  return " ".join([lemmatizer.lemmatize(word) for word in text.split()])
```

```
df['reviewText']=df['reviewText'].apply(lambda x:lemmatize_words(x))
```

```
<ipython-input-25-7fa46fdadeba>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
  df['reviewText']=df['reviewText'].apply(lambda x:lemmatize_words(x))
```

```
df.head()
```

	reviewText	rating
0	jace rankin may short he nothing mess man haul...	1
1	great short read didnt want put read one sitti...	1
2	ill start saying first four book wasnt expecti...	1
3	aggie angela lansbury carry pocketbook instead...	1
4	expect type book library pleased find price right	1

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
## Train test split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df['reviewText'],df['rating'], test_size=0.20, random_state=42)
```

```
## text to vector
from sklearn.feature_extraction.text import CountVectorizer
bow = CountVectorizer()
X_train_bow = bow.fit_transform(X_train).toarray()
X_test_bow = bow.transform(X_test).toarray()
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer()
X_train_tfidf = tfidf.fit_transform(X_train).toarray()
X_test_tfidf = tfidf.transform(X_test).toarray()

```

```
X_train_bow
```

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

```
from sklearn.naive_bayes import GaussianNB
nv_model_bow = GaussianNB().fit(X_train_bow, y_train)
nv_model_tfidf = GaussianNB().fit(X_train_tfidf, y_train)

```

```
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

```

```
y_pred_bow = nv_model_bow.predict(X_test_bow)
y_pred_tfidf = nv_model_tfidf.predict(X_test_tfidf)
```

```
confusion_matrix(y_test, y_pred_bow)
```

```
array([[499, 304],
       [717, 880]])
```

```
print("BOW Accuracy", accuracy_score(y_test, y_pred_bow))
```

```
BOW Accuracy 0.5745833333333333
```

```
confusion_matrix(y_test, y_pred_tfidf)
```

```
array([[488, 315],
       [696, 901]])
```

```
print("TFIDF Accuracy", accuracy_score(y_test, y_pred_tfidf))
```

```
TFIDF Accuracy 0.57875
```

```
## Word2Vec
import gensim
```

Double-click (or enter) to edit

```
from nltk.tokenize import word_tokenize

# Tokenizing train and test data
X_train_tokens = [word_tokenize(sent.lower()) for sent in X_train]
X_test_tokens = [word_tokenize(sent.lower()) for sent in X_test]
```

```
from gensim.models import Word2Vec, KeyedVectors
```

```
import gensim.downloader as api
wv = api.load('word2vec-google-news-300')
```

```
[==================================================] 100.0% 1662.8/1662.8MB downloaded
```

```
from gensim.models import Word2Vec

# Train Word2Vec on training data only
w2v_model = Word2Vec(sentences=X_train_tokens, vector_size=100, window=5, min_count=1, workers=4)

```

```
import numpy as np
def sentence_vector(sentence, model, vector_size=100):
    vectors = [model.wv[word] for word in sentence if word in model.wv]
    return np.mean(vectors, axis=0) if vectors else np.zeros(vector_size)

# Convert train and test sets separately
X_train_vectors = np.array([sentence_vector(sent, w2v_model) for sent in X_train_tokens])
X_test_vectors = np.array([sentence_vector(sent, w2v_model) for sent in X_test_tokens])

```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

```

```
# Train Random Forest Classifier
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train_vectors, y_train)
```

```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
# Predict on test data
y_pred = rf_model.predict(X_test_vectors)

```

```
# Evaluate model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

```
Accuracy: 0.76
```

Generated code may be subject to a licence | djangotraining/Project

```
confusion_matrix(y_test, y_pred)
```

```
array([[ 411,  392],
       [ 192, 1405]])
```

```
classification_report(y_test, y_pred)
```

```
'              precision    recall  f1-score   support\n\n           0       0.68      0.51      0.58       803\n           1
0.78      0.88      0.83      1597\n\n    accuracy                           0.76      2400\n   macro avg       0.73      0.70
0.71      2400\nweighted avg       0.75      0.76      0.75      2400\n'
```

1 Start coding or generate with AI.