Machine Learning (classification problem) Exam question

Problem Statement

In this task, you are provided with a dataset that includes several features,

`reviews` this column has keywords of the review

`rating` this column has reviews of the app

`Source` this column has store the app was downloaded from

`Country_code` this column indicate to which country the reviewer is from

Your task is to clean, visualize and apply a machine learning model to accurately predict this outcome and evaluate the performance of your model. Your targted column is the `review` column.

Step 1: Import necessary libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matr
```

Step 2: Read the Dataset

```
# Load the dataset
df = pd.read_csv('Exam_reviews.csv')
df.head()
```

	reviews	rating	source	country_code
0	['cluttered', 'confusing', 'put', 'back']	3		
1	['blood', 'work', 'impossible', 'understand', ...	2		
2	['love', 'mychart', 'yrs', 'access', 'records'...	5		
3	['enjoy', 'using', 'mychart', 'easy', 'use', '...	5		

country_code

الرقم الدولى

Next steps: Generate code with df View recommended plots

Step 3: Explore the dataset

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4689 entries, 0 to 4688
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   reviews       4689 non-null   object
 1   rating        4689 non-null   int64 
 2   source        4689 non-null   object
 3   country_code  4689 non-null   object
dtypes: int64(1), object(3)
memory usage: 146.7+ KB
```

```
# Check for missing values
df.isnull().sum()
```

```
reviews         0
rating          0
source          0
country_code    0
dtype: int64
```

```
df.describe()
```

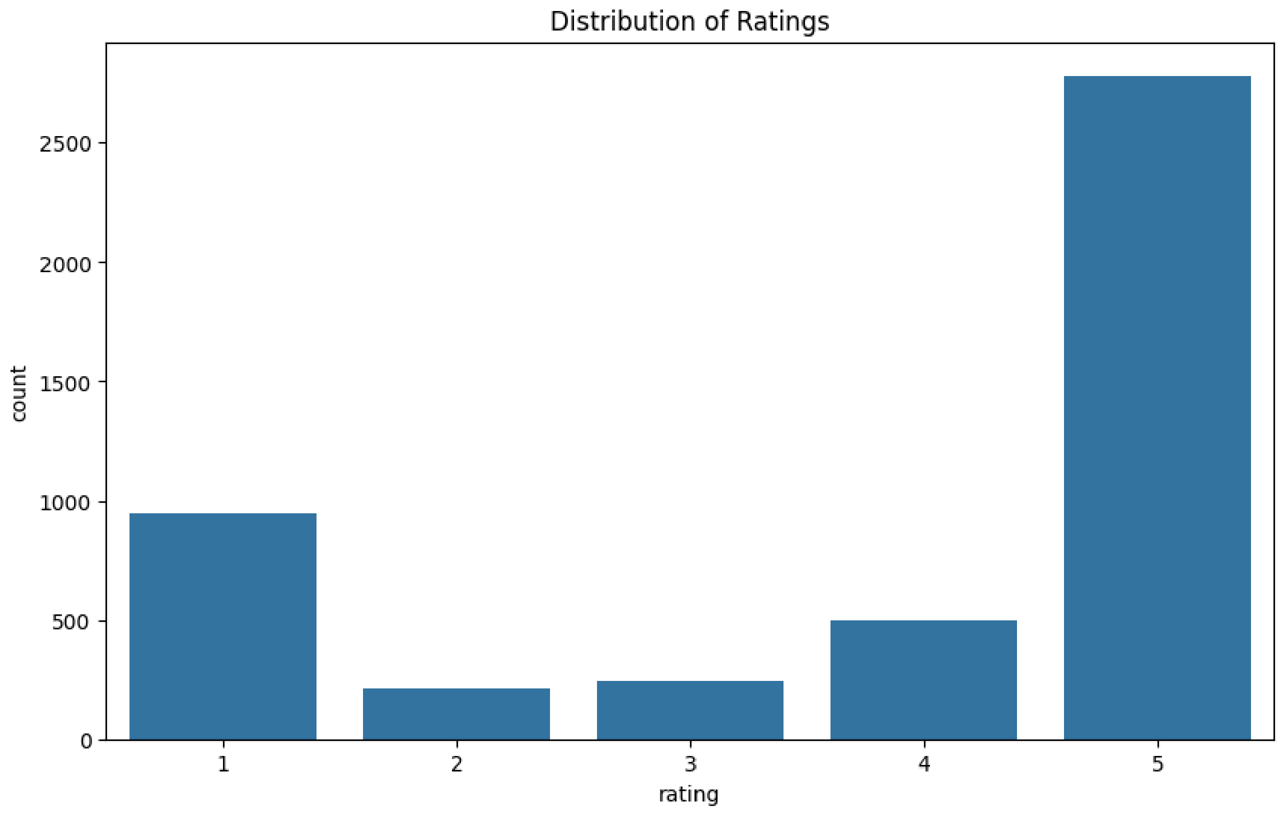
	rating
count	4689.000000
mean	3.843890
std	1.618982
min	1.000000
25%	3.000000
50%	5.000000
75%	5.000000
max	5.000000

Step 4: preprocess the dataset (if needed)

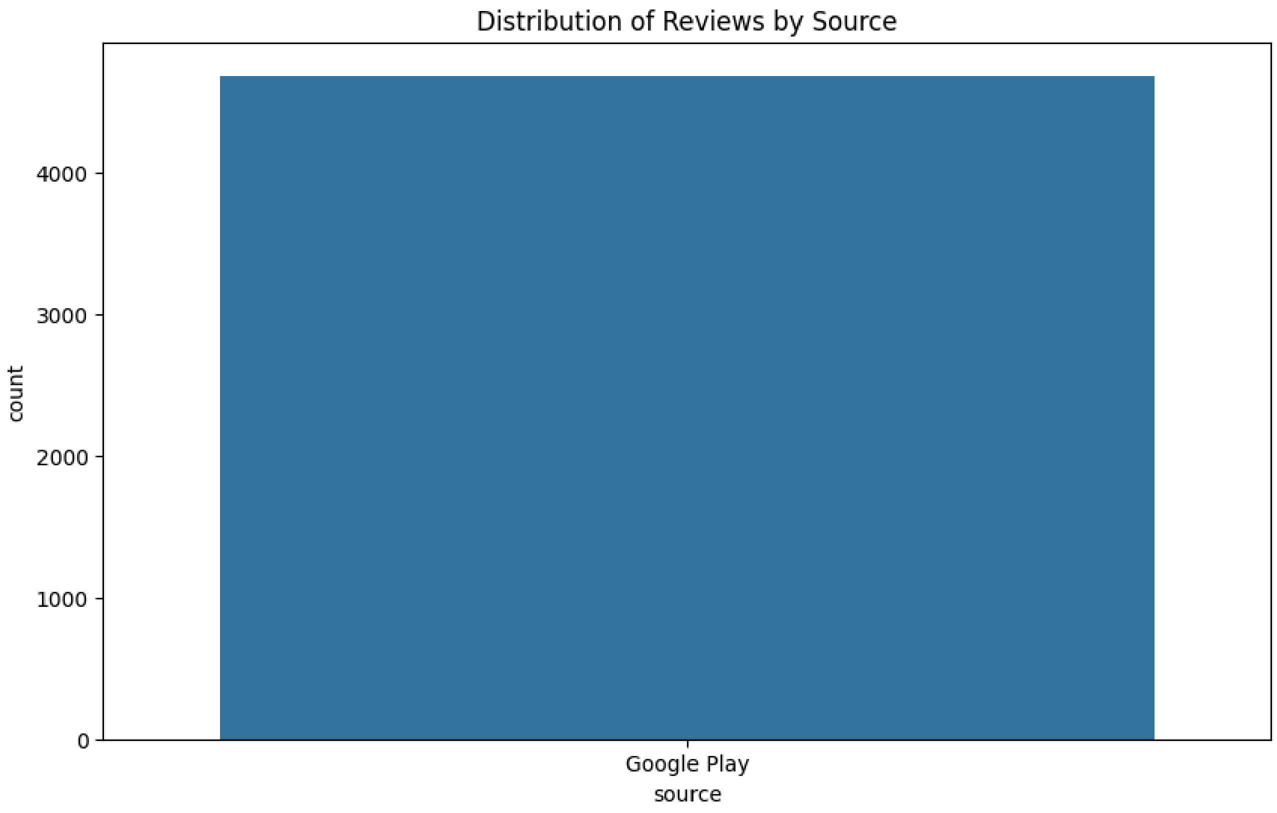
```
# Drop any missing values
df.dropna(inplace=True)

# Example preprocessing step for text data
vectorizer = CountVectorizer(stop_words='english')
X = vectorizer.fit_transform(df['reviews'])

# Define the target variable
y = df['rating']
```

Step 5: Visualize and see the relations between the features

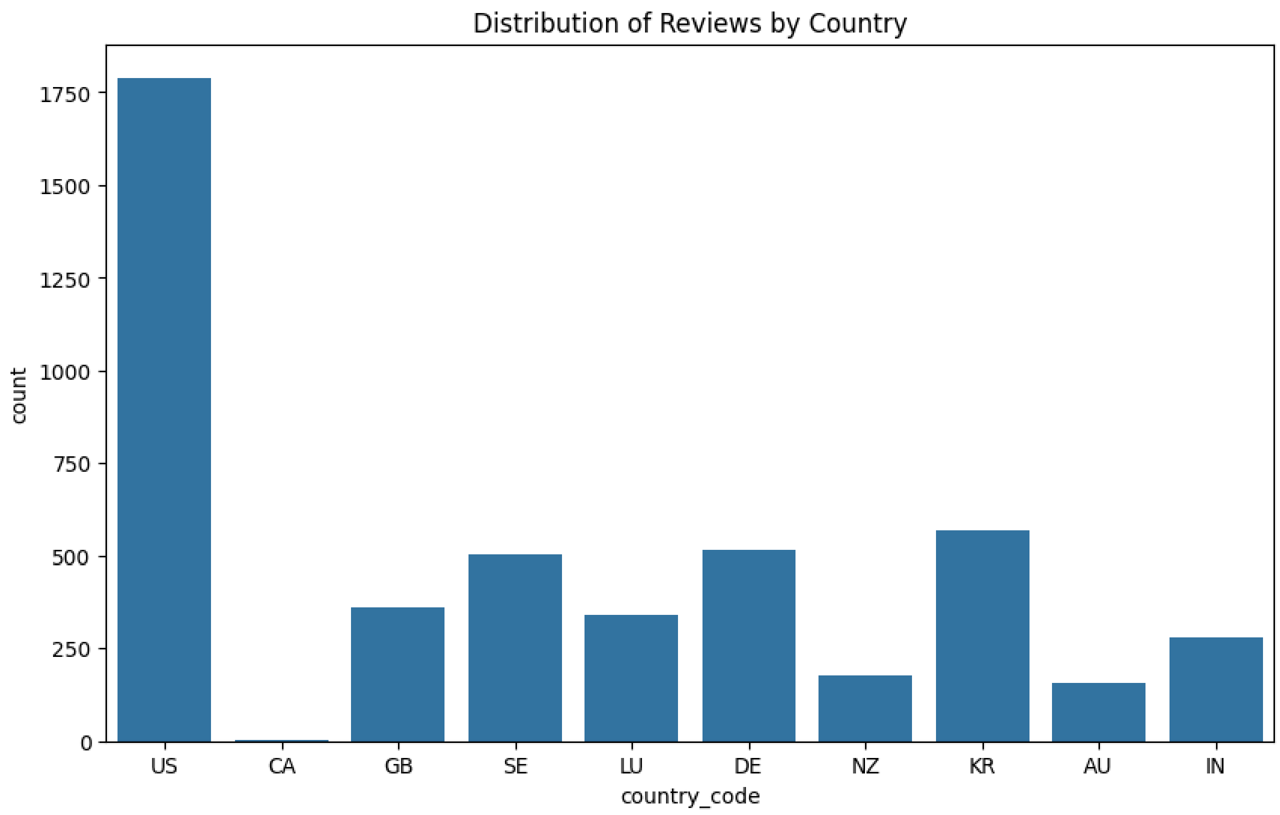
```
# Visualize the distribution of ratings
plt.figure(figsize=(10, 6))
sns.countplot(x='rating', data=df)
plt.title('Distribution of Ratings')
plt.show()
```



```
# Visualize the distribution of reviews by source
plt.figure(figsize=(10, 6))
sns.countplot(x='source', data=df)
plt.title('Distribution of Reviews by Source')
plt.show()
```



```
# Visualize the distribution of reviews by country
plt.figure(figsize=(10, 6))
sns.countplot(x='country_code', data=df)
plt.title('Distribution of Reviews by Country')
plt.show()
```



Step 6: split the dataset and start training

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_st

# Train the model
model = MultinomialNB()
model.fit(X_train, y_train)
```

```
▾ MultinomialNB
MultinomialNB()
```

Step 7: use cross-validation and Test your model

```
# Cross-validation
cv_scores = cross_val_score(model, X_train, y_train, cv=5)
print("Cross-validation scores:", cv_scores)
print("Mean cross-validation score:", np.mean(cv_scores))
```

```
Cross-validation scores: [0.74567244 0.72133333 0.71466667 0.73066667 0.73333
Mean cross-validation score: 0.7291344873501997
```

```
# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Classification report
print("Classification Report:\n", classification_report(y_test, y_pred))
```