Project ID - #CC69849

Project Title - Analyze sentiment in movie reviews

Internship Domain - Data Science Intern

Project Level - Entry Level

Assigned By- Code Clause Internship

Name: Pooja K

Project Details

Aim

Apply K-Means clustering to segment customers based on their purchase behavior.

Description

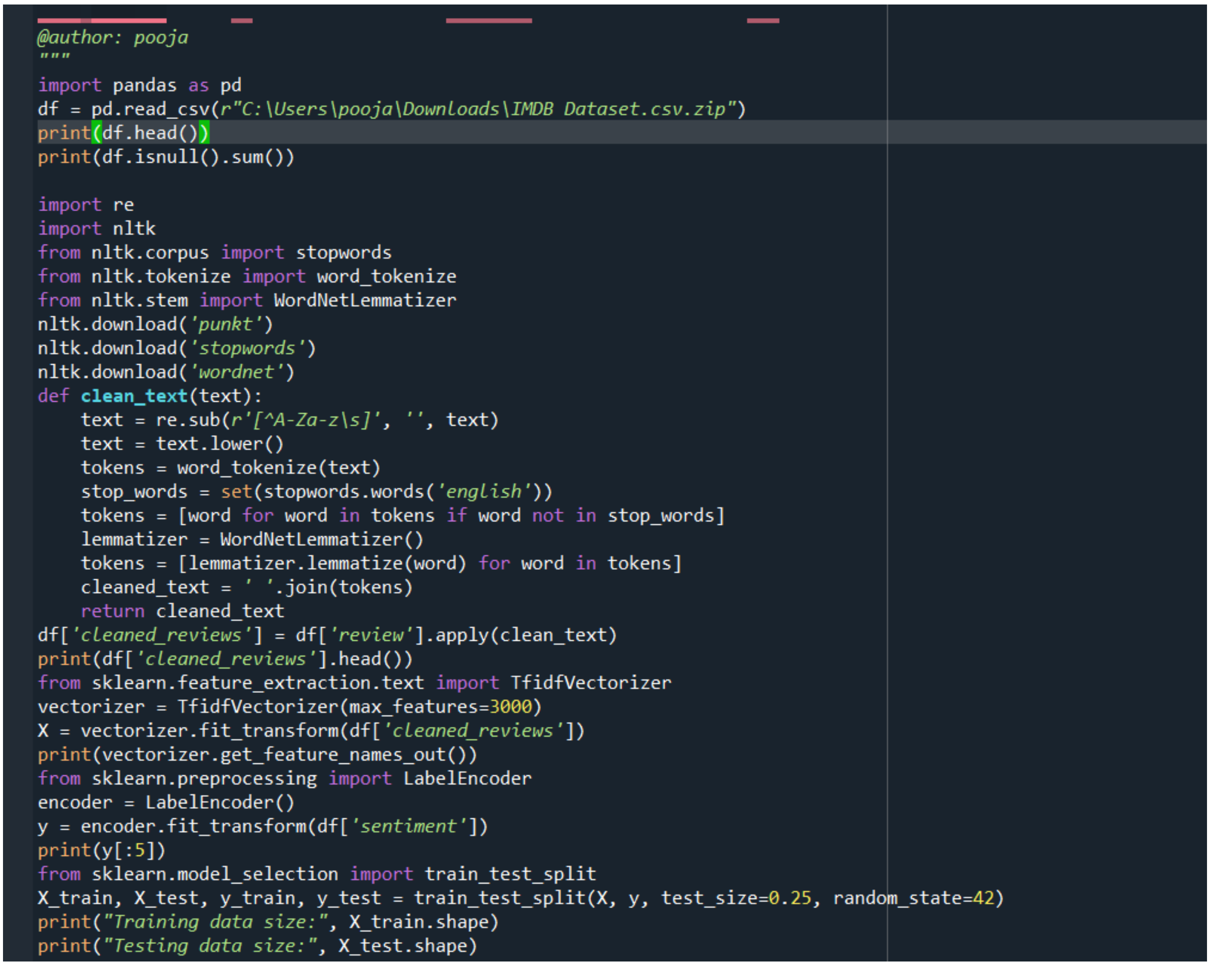
Use Natural Language Processing (NLP) techniques to preprocess text data and build a sentiment analysis model.

Technologies

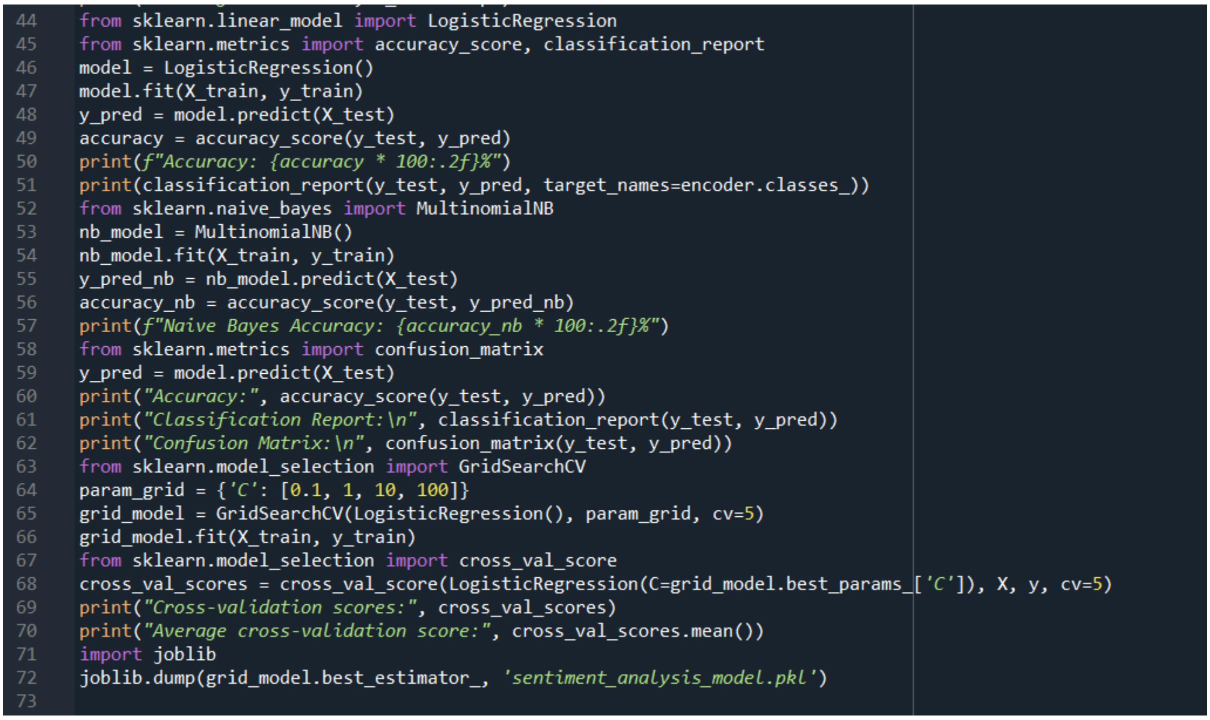
Python, Pandas, NLTK.

You can use other technologies that you know.

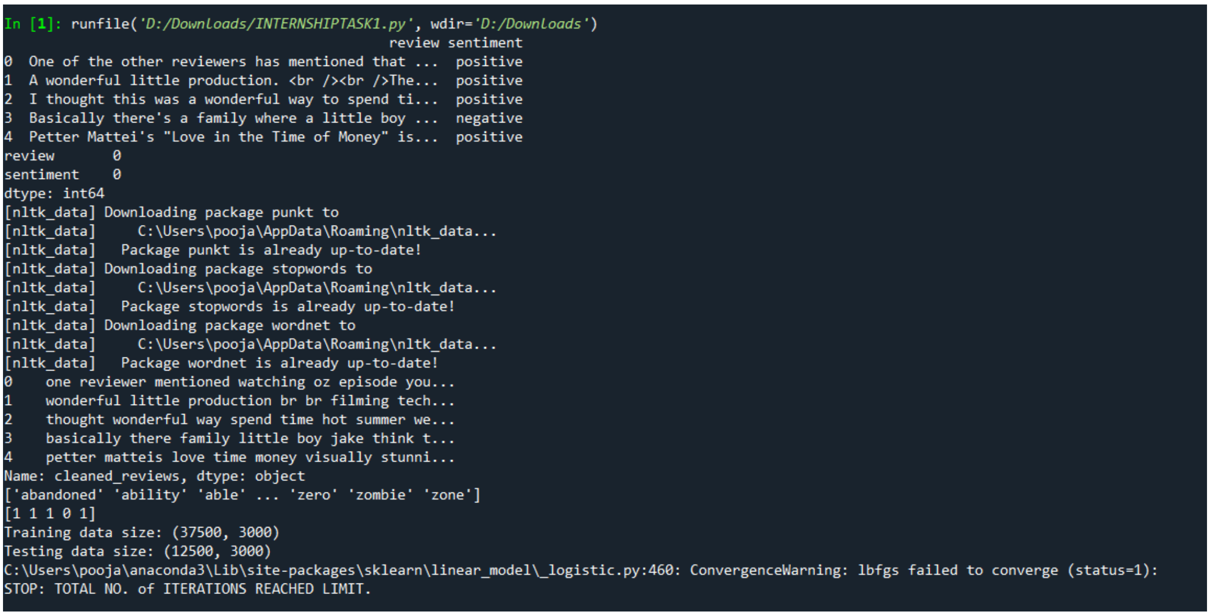
Code



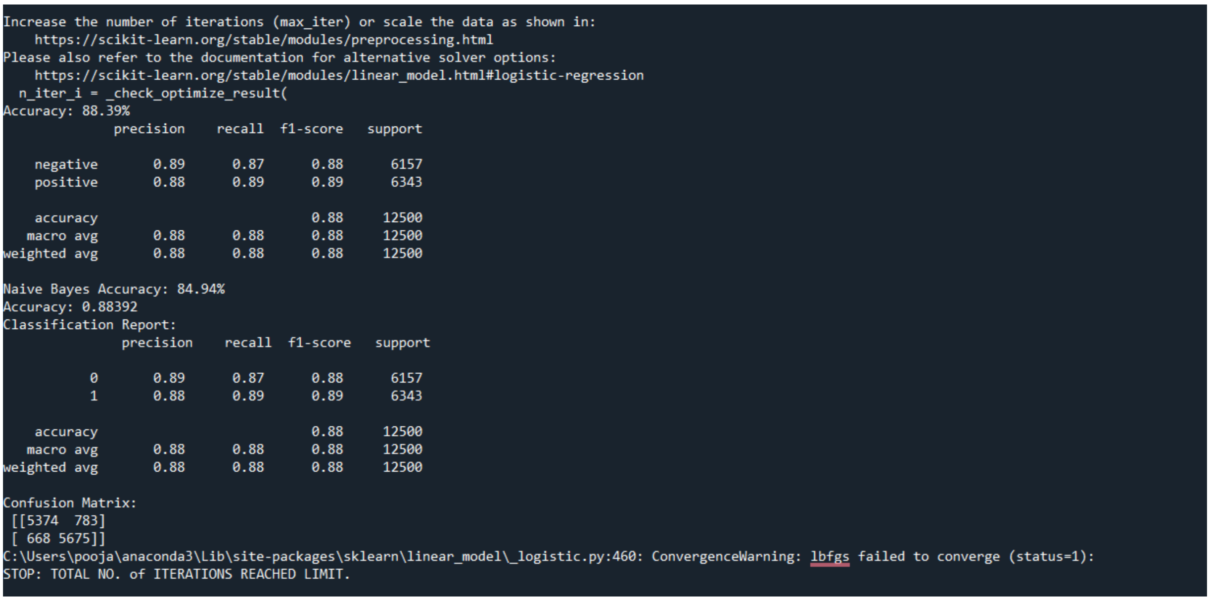
```
@author: pooja
"""
import pandas as pd
df = pd.read_csv(r"C:\Users\pooja\Downloads\IMDB Dataset.csv.zip")
print(df.head())
print(df.isnull().sum())

import re
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
def clean_text(text):
    text = re.sub(r'[^A-Za-z\s]', '', text)
    text = text.lower()
    tokens = word_tokenize(text)
    stop_words = set(stopwords.words('english'))
    tokens = [word for word in tokens if word not in stop_words]
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(word) for word in tokens]
    cleaned_text = ' '.join(tokens)
    return cleaned_text
df['cleaned_reviews'] = df['review'].apply(clean_text)
print(df['cleaned_reviews'].head())
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(max_features=3000)
X = vectorizer.fit_transform(df['cleaned_reviews'])
print(vectorizer.get_feature_names_out())
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
y = encoder.fit_transform(df['sentiment'])
print(y[:5])
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
print("Training data size:", X_train.shape)
print("Testing data size:", X_test.shape)
```



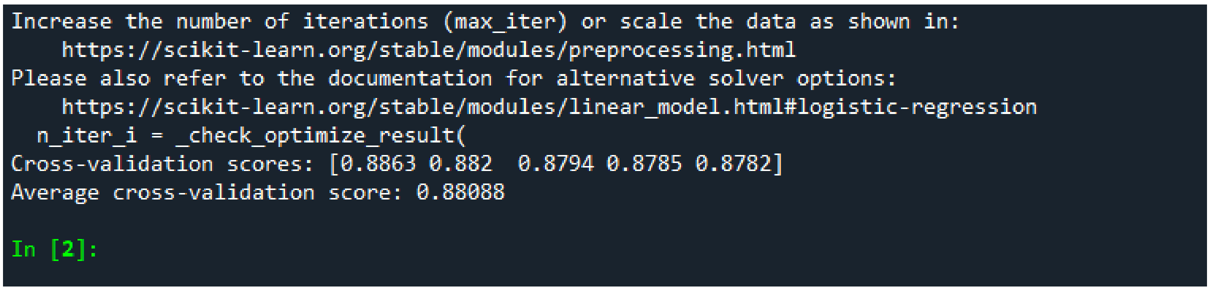
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")
print(classification_report(y_test, y_pred, target_names=encoder.classes_))
from sklearn.naive_bayes import MultinomialNB
nb_model = MultinomialNB()
nb_model.fit(X_train, y_train)
y_pred_nb = nb_model.predict(X_test)
accuracy_nb = accuracy_score(y_test, y_pred_nb)
print(f"Naive Bayes Accuracy: {accuracy_nb * 100:.2f}%")
from sklearn.metrics import confusion_matrix
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
from sklearn.model_selection import GridSearchCV
param_grid = {'C': [0.1, 1, 10, 100]}
grid_model = GridSearchCV(LogisticRegression(), param_grid, cv=5)
grid_model.fit(X_train, y_train)
from sklearn.model_selection import cross_val_score
cross_val_scores = cross_val_score(LogisticRegression(C=grid_model.best_params_['C']), X, y, cv=5)
print("Cross-validation scores:", cross_val_scores)
print("Average cross-validation score:", cross_val_scores.mean())
import joblib
joblib.dump(grid_model.best_estimator_, 'sentiment_analysis_model.pkl')
```



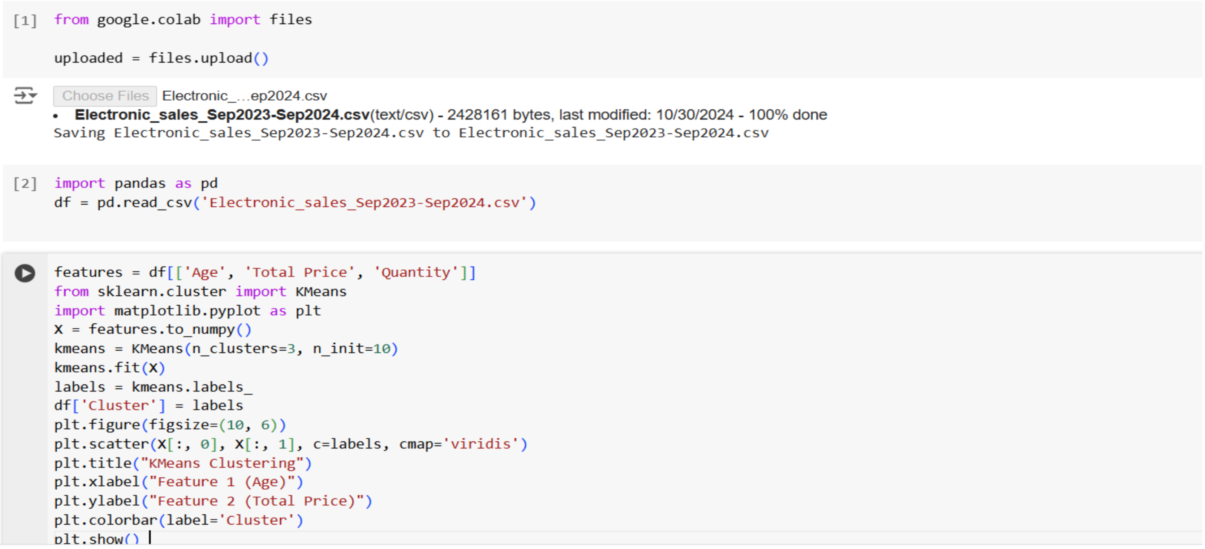
```
In [1]: runfile('D:/Downloads/INTERNSHIPTASK1.py', wdir='D:/Downloads')
                                              review sentiment
0  One of the other reviewers has mentioned that ...  positive
1  A wonderful little production. <br /><br />The...  positive
2  I thought this was a wonderful way to spend ti...  positive
3  Basically there's a family where a little boy ...  negative
4  Petter Mattei's "Love in the Time of Money" is...  positive
review       0
sentiment    0
dtype: int64
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\pooja\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\pooja\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\pooja\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
0    one reviewer mentioned watching oz episode you...
1    wonderful little production br br filming tech...
2    thought wonderful way spend time hot summer we...
3    basically there family little boy jake think t...
4    petter matteis love time money visually stunni...
Name: cleaned_reviews, dtype: object
['abandoned' 'ability' 'able' ... 'zero' 'zombie' 'zone']
[1 1 1 0 1]
Training data size: (37500, 3000)
Testing data size: (12500, 3000)
C:\Users\pooja\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:460: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```



```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
Accuracy: 88.39%
              precision    recall  f1-score   support

    negative       0.89      0.87      0.88      6157
    positive       0.88      0.89      0.89      6343

    accuracy                           0.88     12500
   macro avg       0.88      0.88      0.88     12500
weighted avg       0.88      0.88      0.88     12500

Naive Bayes Accuracy: 84.94%
Accuracy: 0.88392
Classification Report:
               precision    recall  f1-score   support

           0       0.89      0.87      0.88      6157
           1       0.88      0.89      0.89      6343

    accuracy                           0.88     12500
   macro avg       0.88      0.88      0.88     12500
weighted avg       0.88      0.88      0.88     12500

Confusion Matrix:
 [[5374  783]
 [ 668 5675]]
C:\Users\pooja\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:460: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```



```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
Cross-validation scores: [0.8863 0.882  0.8794 0.8785 0.8782]
Average cross-validation score: 0.88088

In [2]:
```

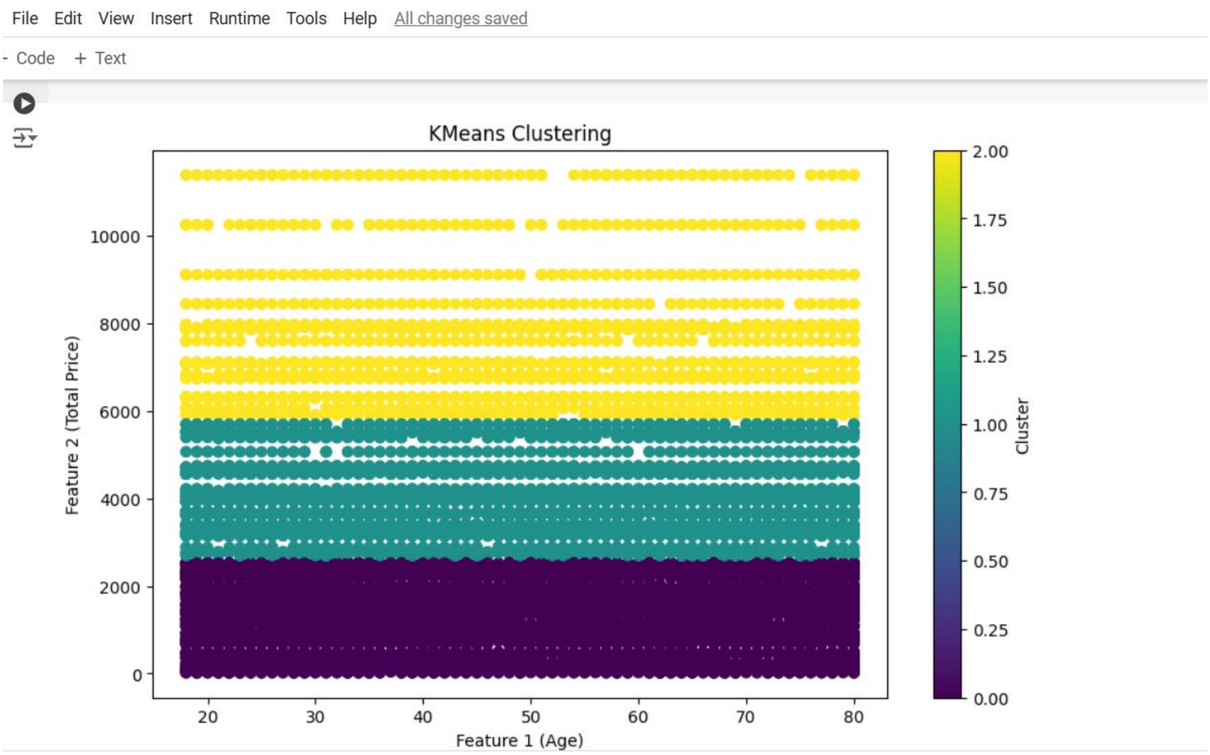


```
[1] from google.colab import files

    uploaded = files.upload()
```

```
Choose Files  Electronic_...ep2024.csv
• Electronic_sales_Sep2023-Sep2024.csv(text/csv) - 2428161 bytes, last modified: 10/30/2024 - 100% done
Saving Electronic_sales_Sep2023-Sep2024.csv to Electronic_sales_Sep2023-Sep2024.csv
```

```
[2] import pandas as pd
    df = pd.read_csv('Electronic_sales_Sep2023-Sep2024.csv')
```

```
features = df[['Age', 'Total Price', 'Quantity']]
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
X = features.to_numpy()
kmeans = KMeans(n_clusters=3, n_init=10)
kmeans.fit(X)
labels = kmeans.labels_
df['Cluster'] = labels
plt.figure(figsize=(10, 6))
plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis')
plt.title("KMeans Clustering")
plt.xlabel("Feature 1 (Age)")
plt.ylabel("Feature 2 (Total Price)")
plt.colorbar(label='Cluster')
plt.show()
```



Conclusion:

The Analyze Sentiment in Movie Reviews project successfully used Natural Language Processing (NLP) and machine learning to derive sentiment insights from movie reviews, ultimately helping us understand customer attitudes toward movies. By applying K-Means clustering, we segmented reviews based on similar sentiment and behaviors, identifying distinct groups of customer sentiment patterns. Preprocessing techniques such as tokenization, stop-word removal, and stemming were instrumental in preparing the data for analysis. The clustering model provided meaningful customer segments, aiding in targeted marketing and recommendation strategies. This project highlights the value of combining NLP and clustering techniques for actionable insights, with potential to enhance customer satisfaction and engagement in entertainment industries. Further improvements could include fine-tuning clusters and exploring deep learning for greater sentiment precision.