```
import pandas as pd
import numpy as np
import csv
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix

# Data
X_train = ["This was really awesome an awesome movie",
           "Great movie! I likes it a lot",
           "Happy Ending! Awesome Acting by hero",
           "loved it!",
           "Bad not upto the mark",
           "Could have been better",
           "really Dissapointed by the movie"]

y_train = ["positive", "positive", "positive", "positive", "negative", "negative",
"negative"]

# Create DataFrame
df_train = pd.DataFrame({"Review": X_train, "Sentiment": y_train})

# Use CountVectorizer
cv = CountVectorizer(ngram_range=(1, 2))
X_vec_train = cv.fit_transform(df_train["Review"]).toarray()

# Train a Naive Bayes classifier
mn = MultinomialNB()
mn.fit(X_vec_train, df_train["Sentiment"])

# Predict on the training data
Y_pred_train = mn.predict(X_vec_train)

# Calculate and print the accuracy on the training data
accuracy_train = accuracy_score(df_train["Sentiment"], Y_pred_train)
print("Accuracy on training data:", accuracy_train)

# Confusion Matrix
conf_matrix = confusion_matrix(df_train["Sentiment"], Y_pred_train,
labels=["positive", "negative"])

# Plot the confusion matrix
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues",
xticklabels=["positive", "negative"], yticklabels=["positive", "negative"])
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title("Confusion Matrix")

# Move plt.show() outside the indented block
plt.show()
```