

1] MOVIE REVIEW SYSTEM

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import csv

# Initialize an empty list to hold chunks
chunks = []

# Read the CSV in chunks with error handling
chunk_size = 5000
for chunk in pd.read_csv("/content/IMDB Dataset.csv", chunksize=chunk_size, encoding='utf-8', quoting=csv.QUOTE_NONE, on_bad_lines='skip'):
    chunks.append(chunk)

# Concatenate all chunks into a single DataFrame
df = pd.concat(chunks, ignore_index=True)

# Ensure correct column names
print(df.columns)

➡ Index(['review', 'sentiment'], dtype='object')

# Handle NaN values in both 'review' and 'sentiment' columns
df = df.dropna(subset=['review', 'sentiment']) # Drop rows where 'review' or 'sentiment' has NaN values

# Split the data into features (X) and labels (y)
X = df['review'] # The text reviews
y = df['sentiment'] # The sentiment (1 or 0)

# Convert text into numerical features using CountVectorizer
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(X)

# Split the dataset into training (70%) and testing (30%) data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Naive Bayes classifier
nb_model = MultinomialNB()
nb_model.fit(X_train, y_train) # Train the model
y_pred_nb = nb_model.predict(X_test) # Predict on the test data
print("Naive Bayes Accuracy:", accuracy_score(y_test, y_pred_nb)) # Print accuracy

➡ Naive Bayes Accuracy: 0.5909090909090909

# Logistic Regression classifier
lr_model = LogisticRegression(max_iter=2000) # Increase max_iter if needed
lr_model.fit(X_train, y_train) # Train the model
y_pred_lr = lr_model.predict(X_test) # Predict on the test data
print("Logistic Regression Accuracy:", accuracy_score(y_test, y_pred_lr)) # Print accuracy

➡ Logistic Regression Accuracy: 0.5454545454545454

```

