Exp7 To perform sentiment analysis on the IMDB review dataset using the Bag of Words (BOW) technique and various machine learning models, you can follow these steps:

**1. Data Preparation**

* Load the text.csv file.
* Split the data into features (text) and labels (label).
* Perform any necessary preprocessing (e.g., removing special characters, lowercasing, tokenization).

**2. Text Vectorization (Bag of Words)**

* Convert the text data into a numerical format using the Bag of Words technique.
* Use CountVectorizer or TfidfVectorizer from sklearn for this purpose.

**3. Model Selection and Training**

* Train various machine learning models on the vectorized data:
  + Logistic Regression
  + Support Vector Machines (SVM)
  + Random Forest
  + Naive Bayes
  + k-Nearest Neighbors (k-NN)
* Split the dataset into training and testing sets.

**4. Model Evaluation**

* Evaluate each model using appropriate metrics (e.g., accuracy, precision, recall, F1 score).

**5. Model Comparison**

* Compare the performance of the models and identify the best one for this task.

I can walk you through the code for each step. If you can provide the text.csv file, I can demonstrate the process directly. Would you like to proceed with this approach?

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.feature\_extraction.text import CountVectorizer, TfidfVectorizer

from sklearn.linear\_model import LogisticRegression

from sklearn.naive\_bayes import MultinomialNB

from sklearn.svm import SVC

from sklearn.ensemble import RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

# Load the dataset

data = pd.read\_csv('Test.csv')

# Inspect the first few rows of the dataset

print(data.head())

# Check for missing values

print(data.isnull().sum())

# Drop missing values (if any)

data.dropna(inplace=True)

# Split the data into features (X) and labels (y)

X = data['text']

y = data['label']

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Vectorization using Bag of Words (BOW)

vectorizer = CountVectorizer()

X\_train\_bow = vectorizer.fit\_transform(X\_train)

X\_test\_bow = vectorizer.transform(X\_test)

# List of models to train

models = {

"Logistic Regression": LogisticRegression(),

"Naive Bayes": MultinomialNB(),

"Support Vector Machine": SVC(),

"Random Forest": RandomForestClassifier(),

"k-NN": KNeighborsClassifier()

}

# Train and evaluate each model

for model\_name, model in models.items():

print(f"\nTraining {model\_name}...")

model.fit(X\_train\_bow, y\_train)

# Predicting the test set results

y\_pred = model.predict(X\_test\_bow)

# Evaluating the model

print(f"\n{model\_name} Accuracy: {accuracy\_score(y\_test, y\_pred):.2f}")

print("Classification Report:")

print(classification\_report(y\_test, y\_pred))

print("Confusion Matrix:")

print(confusion\_matrix(y\_test, y\_pred))

