

```

In [1]: import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline
import random

# Initialize an empty task list
tasks = pd.DataFrame(columns=['description', 'priority'])

# Load pre-existing tasks from a CSV file (if any)
try:
    tasks = pd.read_csv(r"C:\Users\kuppi\Downloads\basic\basic\tasks.csv")
except FileNotFoundError:
    pass

# Function to save tasks to a CSV file
def save_tasks():
    tasks.to_csv('tasks.csv', index=False)

# Train the task priority classifier
vectorizer = CountVectorizer()
clf = MultinomialNB()
model = make_pipeline(vectorizer, clf)
model.fit(tasks['description'], tasks['priority'])

# Function to add a task to the List
# Function to add a task to the List
# Function to add a task to the List
def add_task(description, priority):
    global tasks # Declare tasks as a global variable
    new_task = pd.DataFrame({'description': [description], 'priority': [priority]})
    tasks = pd.concat([tasks, new_task], ignore_index=True)
    save_tasks()

# Function to remove a task by description
def remove_task(description):
    tasks = tasks[tasks['description'] != description]
    save_tasks()

# Function to list all tasks
def list_tasks():
    if tasks.empty:
        print("No tasks available.")
    else:
        print(tasks)

# Function to recommend a task based on machine Learning
# Function to recommend a task based on machine Learning
def recommend_task():
    if not tasks.empty:
        # Get high-priority tasks
        high_priority_tasks = tasks[tasks['priority'] == 'High']

        if not high_priority_tasks.empty:
            # Choose a random high-priority task
            random_task = random.choice(high_priority_tasks['description'])
            print(f"Recommended task: {random_task} - Priority: High")
        else:
            print("No high-priority tasks available for recommendation.")
    else:
        print("No tasks available for recommendations.")

```

4

```
# Main menu
while True:
    print("\nTask Management App")
    print("1. Add Task")
    print("2. Remove Task")
    print("3. List Tasks")
    print("4. Recommend Task")
    print("5. Exit")

    choice = input("Select an option: ")

    if choice == "1":
        description = input("Enter task description: ")
        priority = input("Enter task priority (Low/Medium/High): ").capitalize()
        add_task(description, priority)
        print("Task added successfully.")

    elif choice == "2":
        description = input("Enter task description to remove: ")
        remove_task(description)
        print("Task removed successfully.")

    elif choice == "3":
        list_tasks()

    elif choice == "4":
        recommend_task()

    elif choice == "5":
        print("Goodbye!")
        break

    else:
        print("Invalid option. Please select a valid option.")
```

Task Management App

1. Add Task

2. Remove Task

3. List Tasks

4. Recommend Task

5. Exit

Select an option: 5

Goodbye!

In [ ]: