

```
In [42]: import pandas as pd
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import pickle
import gzip
import re
```

```
In [1]: # Function to Load the saved model
def load_model(model_filename):
    with gzip.open(model_filename, 'rb') as model_file:
        cv, similarity, df = pickle.load(model_file)
    return cv, similarity, df
```

```
In [44]: # Function to preprocess input and get recommendations for both Allopathy and Ayurveda
def recommend_medicines(input_text, allopathy_cv, allopathy_similarity, allopathy_df,
                        ps = PorterStemmer()):
    keywords = [ps.stem(word.lower()) for word in re.findall(r'\b\w+\b', input_text)]

    # Allopathy recommendations
    allopathy_input_vector = allopathy_cv.transform([" ".join(keywords)]).toarray()
    allopathy_input_similarity = cosine_similarity(allopathy_input_vector, allopathy_cv.toarray())

    allopathy_recommendations = []
    for i in range(3):
        index = allopathy_input_similarity.argsort()[0][-i-2]
        allopathy_recommendations.append(allopathy_df.iloc[index]['Drug_Name']) # Replace 'Drug_Name' with actual column name

    # Remove duplicates from the list
    unique_allopathy_recommendations = list(set(allopathy_recommendations))

    # Ayurveda recommendations
    ayurveda_input_vector = ayurveda_cv.transform([" ".join(keywords)]).toarray()
    ayurveda_input_similarity = cosine_similarity(ayurveda_input_vector, ayurveda_cv.toarray())

    ayurveda_recommendations = []
    for i in range(3):
        index = ayurveda_input_similarity.argsort()[0][-i-2]
        ayurveda_recommendations.append(ayurveda_df.iloc[index]['drug']) # Replace 'drug' with actual column name

    # Remove duplicates from the list
    unique_ayurveda_recommendations = list(set(ayurveda_recommendations))

    return unique_allopathy_recommendations, unique_ayurveda_recommendations
```

```
In [45]: # Load Allopathy model
allopathy_cv, allopathy_similarity, allopathy_df = load_model('allopathy_cosine_similarity.pkl')

# Load Ayurveda model
ayurveda_cv, ayurveda_similarity, ayurveda_df = load_model('ayurvedic_cosine_similarity.pkl')
```

```
In [46]: # User input and recommendations for both Allopathy and Ayurveda
user_input_text = input("Enter your symptoms or description: ")
allopathy_recommendations, ayurveda_recommendations = recommend_medicines(
    user_input_text, allopathy_cv, allopathy_similarity, allopathy_df, ayurveda_cv, ayurveda_similarity, ayurveda_df
)
```

```
# Print Allopathy Recommendations
print("\nAllopathy Recommendations:")
for recommendation in allopathy_recommendations:
    print(recommendation)

# Print Ayurveda Recommendations
print("\nAyurveda Recommendations:")
for recommendation in ayurveda_recommendations:
    print(recommendation)
```

Enter your symptoms or description: fever

Allopathy Recommendations:

Calpol 100mg Drops 15mlCalpol 650mg Tablet 10'SCalpol 500mg Tablet 500'SCalpol 500mg
Tablet 10'S

Coldmine Syrup 60ml

Babygesic 250mg Syrup 60mlBabygesic 125mg Syrup 60ml

Ayurveda Recommendations:

phadke

ashta choornam

panchkol churna

In []:

In []:

In []:

In []: