

```
In [4]: import numpy as np # Linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

```
In [5]: medicines = pd.read_csv('./datasets/medicine.csv')
```

```
In [6]: medicines.head()
```

```
Out[6]:
```

	index	Drug_Name	Reason	Description
0	1	A CN Gel(Topical) 20gmA CN Soap 75gm	Acne	Mild to moderate acne (spots)
1	2	A Ret 0.05% Gel 20gmA Ret 0.1% Gel 20gmA Ret 0...	Acne	A RET 0.025% is a prescription medicine that i...
2	3	ACGEL CL NANO Gel 15gm	Acne	It is used to treat acne vulgaris in people 12...
3	4	ACGEL NANO Gel 15gm	Acne	It is used to treat acne vulgaris in people 12...
4	5	Acleen 1% Lotion 25ml	Acne	treat the most severe form of acne (nodular ac...

```
In [7]: medicines.shape
```

```
Out[7]: (9720, 4)
```

```
In [8]: medicines.isnull().sum()
```

```
Out[8]: index      0
Drug_Name    0
Reason       0
Description   0
dtype: int64
```

```
In [9]: medicines.dropna(inplace=True)
```

```
In [10]: medicines.duplicated().sum()
```

```
Out[10]: 0
```

```
In [11]: medicines['Description']
```

```
Out[11]: 0          Mild to moderate acne (spots)
1      A RET 0.025% is a prescription medicine that i...
2      It is used to treat acne vulgaris in people 12...
3      It is used to treat acne vulgaris in people 12...
4      treat the most severe form of acne (nodular ac...
...
9715          used for treating warts
9716      used to soften the skin cells
9717          used for scars
9718          used for wounds
9719      used to treat and remove raised warts (usually...
Name: Description, Length: 9720, dtype: object
```

```
In [12]: medicines['Description'].apply(lambda x:x.split())
```

```
Out[12]: 0          [Mild, to, moderate, acne, (spots)]
1      [A, RET, 0.025%, is, a, prescription, medicine...
2      [It, is, used, to, treat, acne, vulgaris, in, ...
3      [It, is, used, to, treat, acne, vulgaris, in, ...
4      [treat, the, most, severe, form, of, acne, (no...

...
9715          [used, for, treating, warts]
9716          [used, to, soften, the, skin, cells]
9717          [used, for, scars]
9718          [used, for, wounds]
9719  [used, to, treat, and, remove, raised, warts, ...
Name: Description, Length: 9720, dtype: object
```

```
In [13]: medicines['Reason'] = medicines['Reason'].apply(lambda x:x.split())
medicines['Description'] = medicines['Description'].apply(lambda x:x.split())
```

```
In [14]: medicines['Description'] = medicines['Description'].apply(lambda x:[i.replace("
```

```
In [16]: medicines['Description'] = medicines['Description'].apply(lambda x:[i.replace("
```

```
In [17]: medicines['tags'] = medicines['Description'] + medicines['Reason']
```

```
In [18]: new_df = medicines[['index', 'Drug_Name', 'tags']]
```

```
In [19]: new_df
```

Out[19]:

	index	Drug_Name	tags
0	1	A CN Gel(Topical) 20gmA CN Soap 75gm	[Mild, to, moderate, acne, (spots), Acne]
1	2	A Ret 0.05% Gel 20gmA Ret 0.1% Gel 20gmA Ret 0...	[A, RET, 0.025%, is, a, prescription, medicine...
2	3	ACGEL CL NANO Gel 15gm	[It, is, used, to, treat, acne, vulgaris, in, ...
3	4	ACGEL NANO Gel 15gm	[It, is, used, to, treat, acne, vulgaris, in, ...
4	5	Acleen 1% Lotion 25ml	[treat, the, most, severe, form, of, acne, (no...
...
9715	9716	T Muce Ointment 5gm	[used, for, treating, warts, Wound]
9716	9717	Wokadine 10% Solution 100mlWokadine Solution 5...	[used, to, soften, the, skin, cells, Wound]
9717	9718	Wokadine M Onit 10gm	[used, for, scars, Wound]
9718	9719	Wound Fix Solution 100ml	[used, for, wounds, Wound]
9719	9720	Wounsol Ointment 15gm	[used, to, treat, and, remove, raised, warts, ...

9720 rows × 3 columns

In [20]:

new_df['tags'].apply(lambda x:" ".join(x))

Out[20]:

0Mild to moderate acne (spots) Acne
1A RET 0.025% is a prescription medicine that i...
2It is used to treat acne vulgaris in people 12...
3It is used to treat acne vulgaris in people 12...
4treat the most severe form of acne (nodular ac...

...
9715used for treating warts Wound
9716used to soften the skin cells Wound
9717used for scars Wound
9718used for wounds Wound
9719used to treat and remove raised warts (usually...
Name: tags, Length: 9720, dtype: object

In [21]:

new_df

Out[21]:

	index	Drug_Name	tags
0	1	A CN Gel(Topical) 20gmA CN Soap 75gm	[Mild, to, moderate, acne, (spots), Acne]
1	2	A Ret 0.05% Gel 20gmA Ret 0.1% Gel 20gmA Ret 0...	[A, RET, 0.025%, is, a, prescription, medicine...
2	3	ACGEL CL NANO Gel 15gm	[It, is, used, to, treat, acne, vulgaris, in, ...]
3	4	ACGEL NANO Gel 15gm	[It, is, used, to, treat, acne, vulgaris, in, ...]
4	5	Acleen 1% Lotion 25ml	[treat, the, most, severe, form, of, acne, (no...
...
9715	9716	T Muce Ointment 5gm	[used, for, treating, warts, Wound]
9716	9717	Wokadine 10% Solution 100mlWokadine Solution 5...	[used, to, soften, the, skin, cells, Wound]
9717	9718	Wokadine M Onit 10gm	[used, for, scars, Wound]
9718	9719	Wound Fix Solution 100ml	[used, for, wounds, Wound]
9719	9720	Wounsol Ointment 15gm	[used, to, treat, and, remove, raised, warts, ...]

9720 rows × 3 columns

In [22]: `new_df['tags'] = new_df['tags'].apply(lambda x:" ".join(x))`

C:\Users\adixi\AppData\Local\Temp\ipykernel_18380\3089450492.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
new_df['tags'] = new_df['tags'].apply(lambda x:" ".join(x))
```

In [23]: `new_df`

Out[23]:

	index	Drug_Name	tags
0	1	A CN Gel(Topical) 20gmA CN Soap 75gm	Mild to moderate acne (spots) Acne
1	2	A Ret 0.05% Gel 20gmA Ret 0.1% Gel 20gmA Ret 0...	A RET 0.025% is a prescription medicine that i...
2	3	ACGEL CL NANO Gel 15gm	It is used to treat acne vulgaris in people 12...
3	4	ACGEL NANO Gel 15gm	It is used to treat acne vulgaris in people 12...
4	5	Acleen 1% Lotion 25ml	treat the most severe form of acne (nodular ac...
...
9715	9716	T Muce Ointment 5gm	used for treating warts Wound
9716	9717	Wokadine 10% Solution 100mlWokadine Solution 5...	used to soften the skin cells Wound
9717	9718	Wokadine M Onit 10gm	used for scars Wound
9718	9719	Wound Fix Solution 100ml	used for wounds Wound
9719	9720	Wounsol Ointment 15gm	used to treat and remove raised warts (usually...

9720 rows × 3 columns

In [24]: `new_df['tags'] = new_df['tags'].apply(lambda x:x.lower())`

C:\Users\adixi\AppData\Local\Temp\ipykernel_18380\3214958533.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
`new_df['tags'] = new_df['tags'].apply(lambda x:x.lower())`

In [25]: `new_df`

Out[25]:

	index	Drug_Name	tags
0	1	A CN Gel(Topical) 20gmA CN Soap 75gm	mild to moderate acne (spots) acne
1	2	A Ret 0.05% Gel 20gmA Ret 0.1% Gel 20gmA Ret 0...	a ret 0.025% is a prescription medicine that i...
2	3	ACGEL CL NANO Gel 15gm	it is used to treat acne vulgaris in people 12...
3	4	ACGEL NANO Gel 15gm	it is used to treat acne vulgaris in people 12...
4	5	Acleen 1% Lotion 25ml	treat the most severe form of acne (nodular ac...
...
9715	9716	T Muce Ointment 5gm	used for treating warts wound
9716	9717	Wokadine 10% Solution 100mlWokadine Solution 5...	used to soften the skin cells wound
9717	9718	Wokadine M Onit 10gm	used for scars wound
9718	9719	Wound Fix Solution 100ml	used for wounds wound
9719	9720	Wounsol Ointment 15gm	used to treat and remove raised warts (usually...

9720 rows × 3 columns

In [26]: `!pip install nltk`

Requirement already satisfied: nltk in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (3.8.1)
 Requirement already satisfied: click in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from nltk) (8.1.7)
 Requirement already satisfied: joblib in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from nltk) (1.3.2)
 Requirement already satisfied: regex>=2021.8.3 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from nltk) (2023.10.3)
 Requirement already satisfied: tqdm in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from nltk) (4.66.1)
 Requirement already satisfied: colorama in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from click->nltk) (0.4.6)

In [27]: `import nltk`In [28]: `from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()`In [29]: `!pip install -U scikit-learn scipy matplotlib`

Requirement already satisfied: scikit-learn in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (1.3.2)

Requirement already satisfied: scipy in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (1.11.3)

Requirement already satisfied: matplotlib in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (3.8.1)

Requirement already satisfied: numpy<2.0,>=1.17.3 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from scikit-learn) (1.26.1)

Requirement already satisfied: joblib>=1.1.1 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from scikit-learn) (1.3.2)

Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from scikit-learn) (3.2.0)

Requirement already satisfied: contourpy>=1.0.1 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (1.2.0)

Requirement already satisfied: cycler>=0.10 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (0.12.1)

Requirement already satisfied: fonttools>=4.22.0 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (4.44.0)

Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (1.4.5)

Requirement already satisfied: packaging>=20.0 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (23.2)

Requirement already satisfied: pillow>=8 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (10.1.0)

Requirement already satisfied: pyparsing>=2.3.1 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (3.1.1)

Requirement already satisfied: python-dateutil>=2.7 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (2.8.2)

Requirement already satisfied: importlib-resources>=3.2.0 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (6.1.0)

Requirement already satisfied: zipp>=3.1.0 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from importlib-resources>=3.2.0->matplotlib) (3.17.0)

Requirement already satisfied: six>=1.5 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)

```
In [31]: from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(stop_words='english',max_features=5000)
```

```
In [32]: def stem(text):
y = []

for i in text.split():
y.append(ps.stem(i))

return " ".join(y)
```

```
In [33]: new_df['tags'] = new_df['tags'].apply(stem)
```

C:\Users\adixi\AppData\Local\Temp\ipykernel_18380\3213734980.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
new_df['tags'] = new_df['tags'].apply(stem)
```

```
In [34]: cv.fit_transform(new_df['tags']).toarray().shape
```

```
Out[34]: (9720, 806)
```

```
In [35]: vectors = cv.fit_transform(new_df['tags']).toarray()
```

```
In [36]: from sklearn.metrics.pairwise import cosine_similarity
```

```
In [37]: cosine_similarity(vectors)
```

```
Out[37]: array([[1.          , 0.25197632, 0.43643578, ..., 0.          , 0.          ,
                0.          ],
               [0.25197632, 1.          , 0.25660012, ..., 0.19245009, 0.1490712 ,
                0.0860663 ],
               [0.43643578, 0.25660012, 1.          , ..., 0.11111111, 0.0860663 ,
                0.0993808 ],
               ...,
               [0.          , 0.19245009, 0.11111111, ..., 1.          , 0.77459667,
                0.2981424 ],
               [0.          , 0.1490712 , 0.0860663 , ..., 0.77459667, 1.          ,
                0.34641016],
               [0.          , 0.0860663 , 0.0993808 , ..., 0.2981424 , 0.34641016,
                1.          ]])
```

```
In [40]: similarity = cosine_similarity(vectors)
```

```
In [41]: similarity[1]
```

```
Out[41]: array([0.25197632, 1.          , 0.25660012, ..., 0.19245009, 0.1490712 ,
                0.0860663 ])
```

```
In [42]: def recommend(medicine):
          medicine_index = new_df[new_df['Drug_Name'] == medicine].index[0]
          distances = similarity[medicine_index]
          medicines_list = sorted(list(enumerate(distances)),reverse=True,key=lambda x:
          medicine_list_recommended=[])

          for i in medicines_list:
              medicine_list_recommended.append(new_df.iloc[i[0]].Drug_Name)
              # print(new_df.iloc[i[0]].Drug_Name)
          return medicine_list_recommended
```

```
In [43]: recommend("Airkast L Tablet 10'S")
```

```
Out[43]: ["Akair LC 10/5mg Tablet 10'S",
          "Aldine 5mg Tablet 10'S",
          'Acnecure Gel 20gm',
          'Acnesol A Nano Gel 15gm',
          'Acnin Cream 50gm']
```

```
In [44]: user_item_interactions = pd.read_csv('./datasets/interactions.csv')
```

```
In [48]: !pip install scikit-surprise
```


Collecting scikit-surprise

Using cached scikit-surprise-1.1.3.tar.gz (771 kB)

Requirement already satisfied: joblib>=1.0.0 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from scikit-surprise) (1.3.2)

Requirement already satisfied: numpy>=1.17.3 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from scikit-surprise) (1.26.1)

Requirement already satisfied: scipy>=1.3.2 in c:\users\adixi\appdata\local\programs\python\python39\lib\site-packages (from scikit-surprise) (1.11.3)

Using legacy 'setup.py install' for scikit-surprise, since package 'wheel' is not installed.

Installing collected packages: scikit-surprise

Running setup.py install for scikit-surprise: started

Running setup.py install for scikit-surprise: finished with status 'done'

Successfully installed scikit-surprise-1.1.3

WARNING: You are using pip version 20.2.3; however, version 23.3.1 is available. You should consider upgrading via the 'c:\users\adixi\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.

```
In [45]: from surprise import Dataset, Reader
from surprise.model_selection import train_test_split
from surprise import SVD # You can choose a different algorithm if needed

# Create a reader object specifying the rating scale (e.g., 1 to 5)
reader = Reader(rating_scale=(1, 5))

# Load the user-item interactions data into a Surprise dataset
data_collab = Dataset.load_from_df(user_item_interactions[['User_ID', 'Drug_Name

# Split the data into train and test sets
trainset, testset = train_test_split(data_collab, test_size=0.2)

# Build and train the collaborative filtering model (SVD in this example)
model_collab = SVD()
model_collab.fit(trainset)
```

Out[45]: <surprise.prediction_algorithms.matrix_factorization.SVD at 0x2bc6f8eec10>

```
In [50]: def collaborative_filtering_recommendations(user_id, n=5):
# Get the list of medicines that the user has not interacted with
medicines_not_interacted = user_item_interactions[~user_item_interactions['D

# Predict ratings for uninteracted medicines
predictions = []
for medicine in medicines_not_interacted:
    predicted_score = model_collab.predict(user_id, medicine).est
    predictions.append((medicine, predicted_score))

# Sort predictions by estimated score and get the top N recommendations
print(model_collab)
print(reader)
print(user_item_interactions)
top_n_recommendations = sorted(predictions, key=lambda x: x[1], reverse=True)

return [medicine for medicine, _ in top_n_recommendations]
```

```
In [47]: def hybrid_recommendations(user_id, medicine_name, n=5):
# Content-based recommendations
content_recommendations = recommend(medicine_name)
# Collaborative filtering recommendations
```

```

collab_recommendations = collaborative_filtering_recommendations(user_id, n)

content_recommendations=list(content_recommendations)
hybrid_recommendations=content_recommendations
if collab_recommendations is not None:
    hybrid_recommendations = content_recommendations + collab_recommendation

# Deduplicate and return the top N recommendations
top_n_recommendations = list(set(hybrid_recommendations))[:n]

return top_n_recommendations

```

```

In [51]: user_id = 6

recommendations = hybrid_recommendations(user_id, "Airkast L Tablet 10'S")
for i in recommendations:
    print(i)

```

<surprise.prediction_algorithms.matrix_factorization.SVD object at 0x000002BC6F8EEC10>

<surprise.reader.Reader object at 0x000002BC6E69DEE0>

	User_ID	Drug_Name \
0	90	Zione O 20/6.25mg Tablet 10'S
1	3	Syncapone 100mg Tablet 10'SSyncapone 50mg Tabl...
2	6	Vitanext Tablet 10'S
3	97	TELMIVAS 80MG TABLET 10'STELMIVAS AM TABLET 10...
4	23	Xykaa 125mg Drops 15mlXykaa 250mg Suspension 60ml
..
95	54	Emgrast M 250mg Injection 1'SEmgrast M 500MCG ...
96	10	Volapride Plus Capsule 10'S
97	53	KAR B 6.25mg Tablet 10'SKAR B 3.125mg Tablet 10'S
98	27	Argex 500mg Capsule 3'S
99	54	CEPIAM TR 8mg Capsule 10's

	Interaction_Score
0	2
1	1
2	2
3	1
4	4
..	...
95	2
96	3
97	4
98	1
99	4

[100 rows x 3 columns]

Acnin Cream 50gm

GB VERT 16mg Tablet 10'SGB Vert Tablet 10'S

Acnesol A Nano Gel 15gm

Amlodac H Tablet 10'SAmlodac H Tablet 15'S

Acinostop 2Gm Injection 1'SAcinostop 1gm Injection 1'S

```

In [49]: import pickle

```

```

In [211... pickle.dump(new_df.to_dict(),open('medicine_dict.pkl','wb'))
pickle.dump(similarity,open('similarity.pkl','wb'))
collab_filtering_data = {
    "reader": reader,

```

```
"model_collab": model_collab,  
"user_item_interactions": user_item_interactions # Assuming user_item_inter  
}  
  
# Save the dictionary to a pickle file  
with open('collab_filtering.pkl', 'wb') as file:  
    pickle.dump(collab_filtering_data, file)
```