Therapy Recommendation System using the Item-Based Collaborative Filtering

Data Mining Course Project 2021

Shaker Mahmud Khandaker
1st Year, Data Science
University of Trento
shaker.khandaker@studenti.unitn.it

ABSTRACT

A physician's well-informed and safe prescribing therapy decisions depend on knowledge of the patient's previous medical condition and the set of therapies they have or have not undertaken. The doctor incorporates this information with clinical knowledge to conclude which therapy will best address the patient's ailments. In this paper, we develop techniques to automate and predict therapy to a patient for a particular condition based on their medical history. This report provides insights about the implementation of the combination of Collaborative Filtering(CB) and Hybrid variety of solution techniques, including nearest neighbor and co-occurrence approaches. We analyze the potential of these approaches in the advancement of the medical sector.

KEYWORDS

Collaborative Filtering, Recommendation System, Python, Data Mining, Nearest Neighbor, Medication Reconciliation

1 INTRODUCTION

Prescribing a therapy to a patient depends on various factors. Each patient has a unique response to the treatments they may receive. These responses rely on a person's age, allergies, and their list of medications. For an effective treatment, these variables make it extremely complex. In order to predict personalized medical therapy, we can use data mining techniques to analyze large datasets of patient history and behavior. This helps us reduce the error and suggest options for the specific patient. This report has developed an approach on Python implementation for utilizing Item-Based Collaborative Filtering, a Recommender System approach (RS), as a methodology for prescribing therapies – precisely tracing and associating patients with a sequence of trials. The method has been developed to assist healthcare professionals in suggesting the subsequent best therapy for a patient to treat a particular condition.

2 RELATED WORK

Building a recommendation system that offers recommendations by matching and searching similar users, their behavior, and generating suggestions that share characteristics with items that users have rated highly is getting more and more popular. For example, Netflix can predict movies or tv shows, Amazon suggests products to their users with a high accuracy. Many systems are already developed using collaborative filtering to automatically detect omissions in medication lists, identifying drugs that the patient can take that are not on the prescription list. In order to build a system that can predict a therapy to a patient while considering the preferences and responses of the treatments on other patients, we have to start with finding similar patients or treatments. The next step would be to predict how successful a therapy is for a patient that was not yet applied to that individual. For that, we have to determine a strategy to connect patients with similarities and how their recovery rate of a therapy for a specific condition affects the possibility of another patient being cured. Also, we have to verify the accuracy of that recovery rate. Collaborative filtering (CF) is a technique for processing information about users and filtering out items in order to make inferences or predictions about the information of similar users or items. CF offers several approaches to make such predictions. We will find the similar patients based on the conditions they have and how they respond to a therapy, i.e, by analyzing their trials. Memory-based collaborative filtering, User-User filtering and Item-Item collaborative filtering are the types of collaborative Recommender System. We will implement the Memory-based collaborative filtering by finding the Patient-Therapy Interaction Matrix and see how a patient responds to it, meaning, how successful the therapy is. Moreover, for new therapies, conditions, users, we will apply Content-based filtering. Finally, we will use Root Mean Square Error (RMSE) and Spearman's correlation to evaluate the accuracy of our result.

3 PROBLEM STATEMENT

At the center of the problem is a dataset of patients. A patient is an entity who has an id, a collection of attributes as ¡name,value¿ pairs. These patients are connected to a couple of entities, a condition and a therapy, both having a set of attributes as ¡name,value¿ pairs. A condition is an illness and physicists have designed therapies to treat them. Conditions can be short term or long term. Doctors suggest therapy when a patient is suffering from a condition. Suggested therapy is a trial which is a tuple ¡t,p,date, params, success¿. Here, t is a therapy, p is a patient, date is the time it was applied, and params are pairs of attribute name-value that describe the therapy. When a patient has a condition, a sequence of therapies are suggested as trials which may or may not cure the condition. There is a parameter success which indicates the percentage of the effectiveness of a therapy for a particular condition. This parameter may depend on the history of a patient's trials. A patient may not have a trial if a condition is detected and not treated yet. We are to design a system to assist therapists suggest an upcoming therapy for a patient for his//her uncured condition. We can summarize that, given the following inputs:

- A set of patients $\mathbf{P}_{\mathbf{set}}$ with a list of conditions and therapies and trials
- A particular patient and his/her medical history **P**
- An uncured condition $\mathbf{C}_{\mathbf{uncured}}$ The output is:
- A therapy **T**

Moreover, we have to evaluate the accuracy of our output **T**.

4 SOLUTION

We are taking an item-based Collaborative Filtering approach to find the interactions between a patient and a therapy because it shows better outcomes as compared to the user-based method as the resemblance between therapies seems to be consistent than the patients. From these interactions we can find the underlying patterns. We have to find the same therapies as a trial that already worked for our target patient. Our solution will follow the below steps:

- Identify the target Patient and his/history
- Find the matched therapies which have the similar success rate as therapies that were applied to the target patient
- Predict the success rate for the same therapies
- Suggest the therapy to the target patient if the forecasted success rate is higher than the threshold

To implement these steps, the algorithm we are designing consists of following parts:

- Patient-Therapy Interaction Matrix
- Calculating Threshold for successful therapies
- Find correlation between global and user specific therapies
- Condition-Therapy interaction matrix
- Nearest neighbor
- Addressing the Cold Start Problem
- Therapy similarity matrix
- Condition similarity matrix

4.1 Patient-Therapy Interaction Matrix

Creation of an interaction matrix between patients and therapy matrix based on the success rate is considered. This matrix can provide an insight on the best therapies ranking that a patient responds well to.

4.2 Calculating Threshold for successful therapies

We are filtering out the trials that have the therapies which were applied less frequently for a condition. Fro that we are calculating a threshold with the following formula:

$$\text{Threshold} = \text{Threshold} - \text{Threshold} * 25/100$$

4.3 Find correlation between global and patient specific therapies

A famous global therapy for a condition is selected and a correlation is calculated with the therapies a patient responds well to.

Pearson correlation coefficient Formula:

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

r = correlation coefficient
x_i = values of the x-variable in a sample
$\bar{x}$ = mean of the values of the x-variable
y_i = values of the y-variable in a sample
$\bar{y}$ = mean of the values of the y-variable

4.4 Condition-Therapy Interaction Matrix

Creation of an interaction matrix between condition and therapy matrix based on the success rate is considered. This matrix can provide an insight on the best therapies ranking for a condition.

4.5 Nearest neighbor

From the Condition-Therapy interaction matrix, we will apply the Euclidean Distance formula for nearest neighbors to calculate the therapy ranking based on success rate for each condition.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

4.6 Addressing the Cold Start Problem

So far, we are only considering the cases where we can find correlations among conditions, therapies and patients who has previous histories. We still need to address the following cold start problems:

- A new patient who has no records
- A patient who has conditions, but no trials
- A new condition

4.7 Therapy similarity matrix

We are applying Content-based filtering to the therapies so that for a particular therapy, we can find similar therapies. So, when there is a new therapy, it will also be considered.

4.8 Condition similarity matrix

We are applying Content-based filtering to the conditions so that for a particular condition, we can find similar conditions. So, when there is a new condition, it will also be considered.

4.9 Select global best therapy to a new patient

For a new patient with no records, we are finding the global best successful therapy for the specific condition and suggest that.

5 IMPLEMENTATION

To implement the above mentioned solution, we have designed the following steps:

- Tuning up dataset and create Trials dataframe
- Analysis of the Dataset
- Fetching ConditionID with Patient ConditionID
- Calculating Global Top Therapies for a Condition
- Calculating mean success rate for treatments
- Calculating Number of Successful Trials
- Creating Patient-Therapy Interaction Matrix

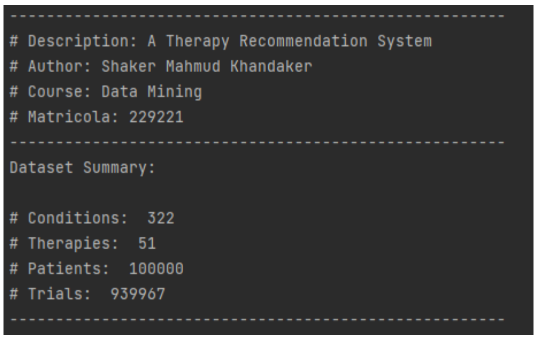
- Making Recommendation for a specific Famous Therapy
- Adding success rate
- Filtering out by selecting most applied therapies

5.1 Tuning up dataset and create dataframe

We are using the following python libraries: json, traceback, datetime, numpy and pandas. In this step, we are taking the Dataset as input and creating two dataframes from Patient Trials called DFtrials and DFconditions.

5.2 Analysis of the Dataset

We are analyzing the dataset to have an overview of the patients, conditions and therapies.

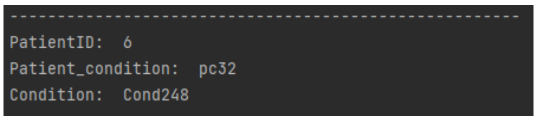


```
---------------------------------------------------
# Description: A Therapy Recommendation System
# Author: Shaker Mahmud Khandaker
# Course: Data Mining
# Matricola: 229221
---------------------------------------------------
Dataset Summary:

# Conditions:  322
# Therapies:  51
# Patients:  100000
# Trials:  939967
---------------------------------------------------
```

Figure 1: Dataset Summary.

5.3 Fetching ConditionID with Patient ConditionID

We are taking our target PatientID and a target Patient ConditionID as input. From this Patient ConditionID, we are fetching the original ConditionID from the DFconditions dataframe.



```
---------------------------------------------------
PatientID:  6
Patient_condition:  pc32
Condition:  Cond248
```

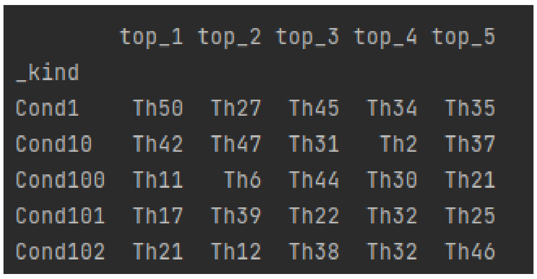
Figure 2: Fetching ConditionID.

5.4 Calculating Global Top Therapies for a Condition

Then we are joining both created dataframes on Condition and PatientID. After that we are pivoting the joined dataframe to get a Condition-Therapy Interaction matrix. For our target Condition, we are extracting the nearest neighbors for the top therapies.

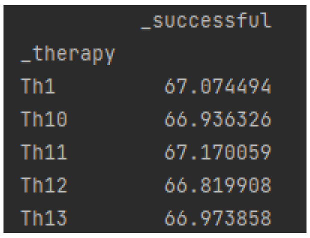
5.5 Calculating mean success rate for treatments

From the joined DFtrial dataframe, we are calculating the mean success rate for therapies by performing a group by operation with therapy and success rate.



```
        top_1 top_2 top_3 top_4 top_5
_kind
Cond1    Th50  Th27  Th45  Th34  Th35
Cond10   Th42  Th47  Th31   Th2  Th37
Cond100  Th11   Th6  Th44  Th30  Th21
Cond101  Th17  Th39  Th22  Th32  Th25
Cond102  Th21  Th12  Th38  Th32  Th46
```

Figure 3: Calculating Global Top Therapies for a Condition.

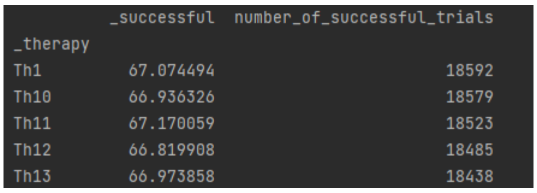


```
          _successful
_therapy
Th1         67.074494
Th10        66.936326
Th11        67.170059
Th12        66.819908
Th13        66.973858
```

Figure 4: Calculating mean success rate for treatments.

5.6 Calculating number of successful trials

From the joined DFtrial dataframe, we are calculating the number of successful therapies by performing a group by operation with therapy and success rate.

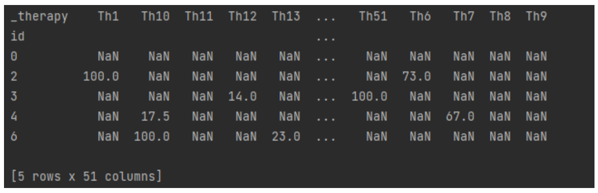


```
          _successful  number_of_successful_trials
_therapy
Th1         67.074494                        18592
Th10        66.936326                        18579
Th11        67.170059                        18523
Th12        66.819908                        18485
Th13        66.973858                        18438
```

Figure 5: Calculating number of successful trials.

5.7 Creating Patient-Therapy Interaction Matrix

We are pivoting the table with PatientID as index and TherapyID as column based on the values of success rate.



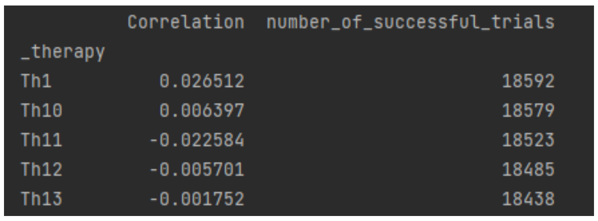
```
_therapy    Th1   Th10  Th11  Th12  Th13  ...   Th51   Th6   Th7  Th8  Th9
id                                        ...
0           NaN    NaN   NaN   NaN   NaN  ...    NaN   NaN   NaN  NaN  NaN
2         100.0    NaN   NaN   NaN   NaN  ...    NaN  73.0   NaN  NaN  NaN
3           NaN    NaN   NaN  14.0   NaN  ...  100.0   NaN   NaN  NaN  NaN
4           NaN   17.5   NaN   NaN   NaN  ...    NaN   NaN  67.0  NaN  NaN
6           NaN  100.0   NaN   NaN  23.0  ...    NaN   NaN   NaN  NaN  NaN

[5 rows x 51 columns]
```

Figure 6: Patient-Therapy Interaction Matrix.

We are also filtering the trials with the count calculated from the previous step with a threshold value. We are considering this threshold value as 25 percent of the successful trials count.

5.8 Making Recommendation for a specific Famous Therapy

From the top therapy calculated in step 5.4, we are making recommendations for the therapies that are best suited for the target patient. We are applying Pearson's correlation coefficient to make this recommendation.



```
          Correlation  number_of_successful_trials
_therapy
Th1          0.026512                        18592
Th10         0.006397                        18579
Th11        -0.022584                        18523
Th12        -0.005701                        18485
Th13        -0.001752                        18438
```

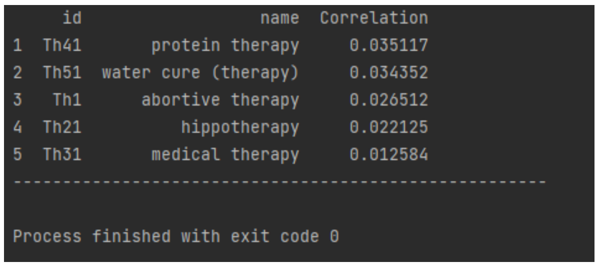
Figure 7: Making Recommendation for a specific Famous Therapy.

5.9 Adding success rate

We are joining the success rate to our recommendation list.

5.10 Filtering out by selecting most applied therapies

We are filtering out the recommendations where the count is less than the calculated threshold value. After this step we are adding the top 5 outputs as therapy suggestions.



```
    id                 name  Correlation
1  Th41       protein therapy     0.035117
2  Th51  water cure (therapy)     0.034352
3   Th1      abortive therapy     0.026512
4  Th21          hippotherapy     0.022125
5  Th31       medical therapy     0.012584
----------------------------------------------------

Process finished with exit code 0
```

Figure 8: Filtering out by selecting most applied therapies.

6 DATASET

In this section, we will discuss the dataset we have used to test and run our system. This dataset is a compilation of health conditions, therapies, patient characteristics and their medical history or trials. This dataset was created by generating random values and connections. We have scrapped several websites for extracting conditions, therapies and dummy patient characteristics. For generating trial histories, we have written a Python Script that creates random data.

6.1 Conditions

```
Conditions : [
    {
        "id": <value>,
        "name": <value>,
        "type": <value>,
        .
        .
    }
]
```

6.2 Therapies

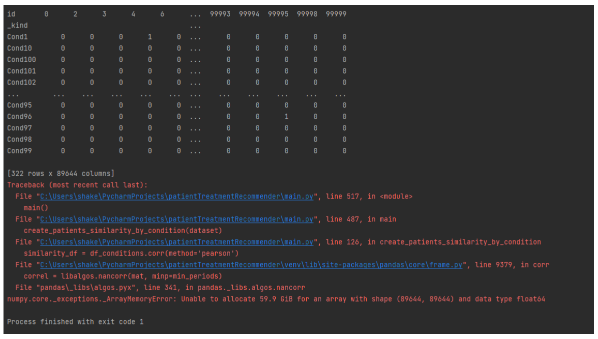
```
Therapies: [
    {
        "id": <value>,
        "name": <value>,
        "type": <value>,
        .
        .
        .
    }
]
```

6.3 Patients

```
Patients: [
    {
        "id": <value>,
        "name": <value>,
        "conditions": [
            {
                "id": <value>,
                "diagnosed": <value>,
                "cured": <value>,
                "kind": <value>,
            },
            .
            .
            .
        ]
        "trials": [
            {
                "id":<value>,
                "start": <value>,
                "end": <value>,
                "condition": <value>,
                "therapy": <value>,
                "successful": <value>,
            },
            .
            .
            .
        ]
    }
]
```

7 EXPERIMENTAL EVALUATION

In this report, instead of evaluating the quality of the predicted therapies, the evaluation of the predicted success rate is performed because it is hard to quantify the effectiveness of a therapy for a condition. We can see that the correlation of the suggested therapies are just about 0.02 when we consider the threshold to 25 percent. We have a failed attempt to create a Patient-Patient similarity matrix but it requires a significant amount of resources which is referred to in **Fig.9**. Regarding computational cost, for a dataset with 322 conditions, 51 therapies, 100000 patients, 939967 trials it takes about six minutes to generate the prediction which is within our expectation.



```
id        0     2     3     4     6    ...  99993  99994  99995  99998  99999
_kind                                  ...
Cond1        0     0     0     1     0 ...      0      0      0      0      0
Cond10       0     0     0     0     0 ...      0      0      0      0      0
Cond100      0     0     0     0     0 ...      0      0      0      0      0
Cond101      0     0     0     0     0 ...      0      0      0      0      0
Cond102      0     0     0     0     0 ...      0      0      0      0      0
...        ...   ...   ...   ...   ... ...    ...    ...    ...    ...    ...
Cond95       0     0     0     0     0 ...      0      0      0      0      0
Cond96       0     0     0     0     0 ...      0      0      1      0      0
Cond97       0     0     0     0     0 ...      0      0      0      0      0
Cond98       0     0     0     0     0 ...      0      0      0      0      0
Cond99       0     0     0     0     0 ...      0      0      0      0      0

[322 rows x 89644 columns]
Traceback (most recent call last):
  File "C:\Users\shake\PycharmProjects\patientTreatmentRecommender\main.py", line 517, in <module>
    main()
  File "C:\Users\shake\PycharmProjects\patientTreatmentRecommender\main.py", line 487, in main
    create_patients_similarity_by_condition(dataset)
  File "C:\Users\shake\PycharmProjects\patientTreatmentRecommender\main.py", line 126, in create_patients_similarity_by_condition
    similarity_df = df_conditions.corr(method='pearson')
  File "C:\Users\shake\PycharmProjects\patientTreatmentRecommender\venv\lib\site-packages\pandas\core\frame.py", line 9379, in corr
    correl = libalgos.nancorr(mat, minp=min_periods)
  File "pandas\_libs\algos.pyx", line 341, in pandas._libs.algos.nancorr
numpy.core._exceptions._ArrayMemoryError: Unable to allocate 59.9 GiB for an array with shape (89644, 89644) and data type float64

Process finished with exit code 1
```

Figure 9: Unsuccessful attemt to run Patient-Patient Similarity matrix.

8 CONCLUSION

The attempt to recommend a therapy to a patient depends on numerous variables. A patient's response to a therapy depends on his demographic attributes as well as his previous trials. Predicting a patient specific therapy is therefore a very sensitive task. To predict therapy, we used CB filtering and to deal with cold start problems, we used CB filtering. The prediction of this system is satisfactory. However, we can improve the performance of our system by implementing clustering and A-priory algorithms.