Develop a personalized medication recommendation system based on patient health records.

A PROJECT REPORT

Submitted by

Aaryan Maheshwari (22BDO10001)

Chayan Gope (22BDO10036)

in partial fulfillment for the award of the degree of

Bachelor of Engineering - Computer Science and Engineering

IN

DevOps

Chandigarh University

April 2024

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

BONAFIDE CERTIFICATE

Certified that this project report **“Develop a personalized medication recommendation system based on patient health records”** is the bonafide work of “**Aaryan Maheshwari (22BDO10001)**, and **Chayan Gope (22BDO10036)”** who carried out the project work under my supervision.

SIGNATURE

Er. Digvijay Puri

HEAD OF THE DEPARTMENT

SIGNATURE

Dr. Ranjan Walia (E14288)

SUPERVISOR

Submitted for the project viva voce examination held on 30 April 2024

INTERNAL EXAMINER

EXTERNAL EXAMINER

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

ACKNOWLEDGEMENT

We would like to express our gratitude and thanks to Dr. Ranjan Walia who took us to the project. His unwavering support and priceless advice served as a lighthouse that guided us forward and ensured the success of this project. We would like to thank Dr. Ranjan Walia who allowed us to do this latest project and provided all the necessary resources. His kind assistance, priceless time, and knowledgeable counsel were indispensable on our trip.

We would also like to thank all these people, especially our friends, whose participation is important in creating a good environment for us. Their participation helps bring new and innovative ideas to the final stage of our project. Their continued support and encouragement are essential; without their help, the program would be daunting. Once again, thanks to all the guides with patience. Dr. Ranjan Walia and our friends worked together. The accomplishment of this job was made possible by their ongoing direction, ongoing assistance, and cooperative efforts.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter – 1

1. INTRODUCTION

1.1. Problem Definition

In the realm of modern healthcare, the pursuit of personalized medicine has emerged as a paramount objective. In this context, our client, a leading healthcare organization, recognizes the pressing need to develop a personalized medication recommendation system based on patient health records. The client's motivation stems from the limitations inherent in traditional healthcare approaches, which often rely on generalized treatment guidelines rather than considering the unique characteristics and needs of individual patients. By harnessing the power of patient health records and cutting-edge technologies, the client seeks to revolutionize healthcare delivery, optimize therapeutic interventions, and enhance patient outcomes.

1.2. Relevant Contemporary Issues

The landscape of healthcare is characterized by several contemporary issues that underscore the urgency and relevance of developing personalized medication recommendation systems. One such issue is the rising prevalence of chronic diseases, which place a significant burden on healthcare systems worldwide. Traditional treatment approaches for chronic conditions often yield suboptimal outcomes due to their failure to account for the diverse genetic, environmental, and lifestyle factors influencing disease progression and treatment response. Additionally, the advent of precision medicine has highlighted the inadequacies of one-size-fits-all treatment paradigms, emphasizing the need for tailored therapeutic interventions that consider individual patient variability.

Another pertinent issue is the exponential growth of health data generated through electronic health records (EHRs), wearable devices, and other digital health technologies. While this wealth of data holds immense potential for improving healthcare outcomes, its sheer volume and complexity present formidable challenges in extracting meaningful insights. Without effective tools and methodologies to analyze and leverage this data, healthcare providers may struggle to deliver personalized care that aligns with patient needs and preferences.

1.3. Problem Identification

Against this backdrop, the primary problem identified by our client is the lack of a robust and scalable system for recommending personalized medications based on patient health records. Existing healthcare IT systems often lack the sophistication to integrate disparate sources of health data, analyze complex patient profiles, and generate tailored treatment recommendations in real time. Consequently, healthcare providers may rely on manual processes and subjective judgment, leading to inconsistencies in treatment decision-making and suboptimal patient outcomes.

1.4. Task Identification

To address this problem, our task is to develop a personalized medication recommendation system that leverages patient health records to provide evidence-based treatment recommendations tailored to individual patients. This system will utilize advanced technologies such as artificial intelligence (AI), machine learning (ML), and natural language processing (NLP) to analyze large volumes of health data, identify patterns and correlations, and generate personalized treatment regimens. Additionally, the system will incorporate features for real-time monitoring, feedback, and adaptation to ensure that treatment recommendations remain dynamic and responsive to changes in patient health status and clinical evidence.

1.5. Timeline

Week 1-2: Topic Selection and Literature Review

- Define the scope of the research and narrow down the focus.
- Conduct an extensive literature review on personalized medication recommendation systems, patient health records, machine learning algorithms, and related topics.
- Identify gaps in existing research and formulate research questions.

Week 3-4: Data Collection and Preprocessing

- Obtain access to relevant patient health records (PHRs) datasets while ensuring compliance with ethical guidelines and data protection laws.
- Clean and preprocess the data to remove noise, handle missing values, and ensure data consistency.
- Explore different data representation techniques suitable for machine learning models.

Week 5 - 6: Integration, Validation, Evaluation and Comparison

- Integrate the developed models with the patient health records system.
- Conduct validation tests to ensure that the recommendation system produces accurate and reliable results.
- Compare the performance of the developed personalized medication recommendation system with existing approaches.
- Evaluate the system's effectiveness in improving patient outcomes, such as medication adherence and health outcomes.
- Analyze the computational efficiency and scalability of the system.

Week 7 - 8: Paper Writing and Submission

- Summarize the research findings, methodology, and results in a research paper format.
- Write the introduction, methodology, results, discussion, and conclusion sections.
- Revise and proofread the paper for clarity, coherence, and adherence to academic standards.
- Submit the paper to relevant conferences or journals for peer review.

Chapter – 2

2. LITERATURE SURVEY

2.1. Existing System

- ***Data-driven Approaches:***

  Data-driven approaches form the backbone of many personalized medication recommendation systems. These systems analyze large volumes of patient health data, including medical history, genetic information, demographic factors, and treatment outcomes, to identify patterns and correlations. Machine learning algorithms, such as decision trees, random forests, and neural networks, are commonly employed to extract insights from this data and generate personalized recommendations. *Example Solution: IBM Watson for Oncology.* IBM Watson for Oncology is a prominent example of a data-driven medication recommendation system. Leveraging artificial intelligence, Watson analyzes vast amounts of medical literature, clinical trial data, and patient records to provide oncologists with personalized treatment recommendations for cancer patients. By considering factors such as tumor genetics, treatment history, and comorbidities, Watson helps clinicians make informed decisions tailored to individual patient needs.

- ***Clinical Decision Support Systems (CDSS):***

  Clinical Decision Support Systems (CDSS) integrate patient health data with clinical knowledge and guidelines to assist healthcare providers in making evidence-based treatment decisions. These systems typically offer real-time recommendations and alerts, helping clinicians navigate complex medical scenarios and optimize patient outcomes. CDSS can encompass a range of functionalities, including medication selection, dosage adjustment, drug-drug interaction detection, and adherence monitoring. *Example Solution: Epic's Clinical Decision Support Tools* Epic, a leading provider of electronic health record (EHR) systems, offers a suite of clinical decision support tools designed to enhance medication management. These tools analyze patient data stored within the EHR, such as lab results, medication history, and allergy information, to deliver context-specific recommendations to prescribers. By integrating seamlessly into the clinician's workflow, Epic's CDSS helps improve medication safety and efficacy while reducing the risk of adverse events.

- ***Pharmacogenomics-based Approaches:***

  Pharmacogenomics-based approaches leverage genetic information to optimize medication selection and dosing for individual patients. By analyzing genetic variations that affect drug metabolism, efficacy, and toxicity, these systems enable personalized prescribing tailored to each patient's unique genetic profile. Pharmacogenomics-based recommendation systems hold particular promise in areas such as psychiatry, where drug response variability is high and genotype-guided prescribing can improve outcomes. *Example Solution: YouScript Precision Prescribing Software*

YouScript Precision Prescribing Software is a pharmacogenomics-based medication recommendation system that analyzes patients' genetic data to guide medication management. By assessing genetic variants in drug-metabolizing enzymes and transporters, YouScript predicts individualized drug responses and identifies potential drug-gene interactions. Healthcare providers can use this information to optimize medication regimens, minimize adverse reactions, and improve therapeutic outcomes for their patients.

- ***Hybrid Approaches:***

Hybrid approaches combine multiple data sources and methodologies to enhance the accuracy and relevance of medication recommendations. These systems integrate clinical data, genetic information, patient preferences, and external knowledge sources to generate comprehensive and personalized treatment plans. By leveraging the strengths of different approaches, hybrid systems aim to deliver more precise and tailored recommendations that account for the multifaceted nature of patient health. *Example Solution: Clinithink's CLiX ENRICH*

Clinithink's CLiX ENRICH is a hybrid medication recommendation system that combines natural language processing (NLP) with clinical decision support capabilities. By analyzing unstructured clinical notes and structured EHR data, CLiX ENRICH extracts valuable insights about patient conditions, treatment histories, and medication preferences. This information is then used to generate personalized medication recommendations that align with evidence-based guidelines and individual patient needs.

2.2. Proposed System

Our proposed system for oral cancer detection using machine learning offers several

advantages over previous approaches, making it a more advanced and effective solution.

Here are some key points highlighting the superiority of our proposed system:

- ***Data Integration and Preprocessing:***

  Our medication system will integrate diverse sources of patient health data, diseases name and it's medications. This data will undergo preprocessing to standardize formats, resolve inconsistencies, and anonymize sensitive information to ensure patient privacy and compliance with regulatory requirements.

- ***Data Analytics and Machine Learning:***

  Our medication system will employ machine learning algorithms to analyze patient health data and identify patterns relevant to medication management. These algorithms will extract features such as medicine to be provided for certain diseases.

- ***Clinical Decision Support:***

  Our medication system will incorporate clinical decision-support capabilities to assist healthcare providers in making evidence-based medication decisions. This includes medication to be provided to the patient for the disease. The system will also provide access to relevant clinical guidelines, literature, and best practices to support informed decision-making.

- ***User Interface and Accessibility:***

  Our medication system will feature a user-friendly interface accessible to healthcare providers across various settings, including hospitals, clinics, and pharmacies. The interface will provide intuitive navigation, customizable dashboards, and interactive visualization tools to facilitate data exploration and decision-making. Integration with existing electronic health record systems will ensure seamless interoperability and workflow integration.

- ***Continuous Learning and Improvement:***

  Our medication system will be designed for continuous learning and improvement through feedback mechanisms and data-driven validation. The system will capture user feedback, clinical outcomes, and real-world data to refine predictive models, update clinical guidelines, and enhance the accuracy of medication recommendations over time. Regular updates and maintenance will ensure that PersonalMedRec remains aligned with the latest advancements in personalized medicine and healthcare informatics.

2.3. Literature Review Summary

Sr.	Year	Objective	Methodology	Advantages	Disadvantages	Future Scope
1.	2021	The objective of the research is to develop an intelligent medication recommendation system that can personalize healthcare by leveraging patient health records.	1. Integration of machine learning with patient health records to tailor medication recommendations based on demographics, medical history, and clinical parameters. 2. Evaluation of effectiveness and accuracy using real-world patient datasets, employing decision trees, support vector machines, and random forests for analysis.	1. Personalized Medication Recommendations: Utilizing patient profiles, the system tailors medication suggestions to enhance treatment effectiveness and patient contentment. 2. Data-Driven Automation: By analyzing extensive health records with machine learning, the system can automate medication selection, lightening the	1. Data Quality: The system's effectiveness hinges on accurate and comprehensive patient health records; incomplete or erroneous data can result in flawed recommendations. 2. Lack of Transparency and Generalization: Machine learning algorithms may generate accurate predictions, but their lack of transparency can hinder	1. Integration of additional data sources like genetic information and wearable device data could enhance medication recommendations. 2. Incorporating patient preferences and treatment goals can personalize healthcare delivery, while continuous model improvement

				workload for healthcare providers and boosting operational efficiency.	understanding, while limitations in generalizing recommendations across diverse patient populations or conditions may arise due to biased or limited training data.	and clinical validation are crucial for real-world effectiveness and adoption.
2.	2020	The objective of the research paper was to develop and evaluate a personalized medication recommendation system that utilizes patient health records to improve treatment outcomes and patient	1. Developed a machine learning model trained on extensive patient health records, extracting pertinent features through feature engineering. 2. Integrated the model into a user-friendly interface for healthcare providers, aiding	1. Enhanced treatment efficacy through personalized medication regimens tailored to individual patient characteristics and medical history. 2. Decreased adverse drug reactions and increased patient satisfaction due to more effective	1. Dependency on thorough patient health records: The personalized medication recommendation system requires extensive and accurate patient health data, which might not consistently exist or be readily obtainable. 2. Interpretability and trust concerns: Healthcare	1. Exploring alternative data sources like wearable devices and genomic data to enhance patient profiles. 2. Incorporating explainable AI techniques to improve model interpretability and

		satisfaction.	in personalized medication regimen predictions during clinical decision-making.	and personalized treatment strategies.	providers faced uncertainty due to the opaque nature of machine learning algorithms, leading to hesitancy in accepting recommendations without insight into the algorithm's decision logic.	trustworthiness, alongside validation studies and clinical trials for real-world effectiveness assessment.
3.	2022	The objective of the research paper is to propose and evaluate a machine learning-based personalized medication recommendat	1. Retrospective analysis of electronic health records (EHRs) from chronic disease patients. 2. Development of personalized medication recommendation models using machine learning	1. Tailored medication recommendations were leveraging machine learning algorithms, aligning treatment with individual patient profiles for enhanced adherence and effectiveness.	1. Electronic health records' quality and completeness vary, impacting reliability across healthcare systems. 2. Challenges include ensuring data privacy, model interpretability, and generalizability to diverse patient	1. Leveraging wearable devices and patient-reported outcomes to refine medication recommendations, ensuring greater precision and individualized

		ion system for patients with chronic diseases, aiming to improve treatment adherence and outcomes.	algorithms like deep neural networks and ensemble methods, evaluated through cross-validation and validation on an independent dataset.	2. Integration of electronic health records enables real-time generation of personalized recommendations, streamlining clinical workflows and mitigating healthcare costs linked to non-adherence and adverse events.	groups.	care. 2. Employing explainable AI methods to enhance transparency in machine learning models, promoting trust among healthcare providers and facilitating the integration of personalized medication recommendations into clinical practice.
4.	2014	The objective of the research was to develop a personalized medication recommendation system	1. Integrated machine learning techniques like decision trees and support vector machines with patient	1. Personalized medication recommendations leverage machine learning to analyze individual patient characteristics,	1. Resource Requirements: Training and deploying machine learning models necessitates significant	1. Integrating genetic information and wearable device data for more precise medication recommendation

		that utilizes patient health records to improve the accuracy and efficacy of medication selection.	health records for personalized medication recommendations. 2. Utilized patient demographics, medical history, lab results, and medication adherence patterns, with data preprocessing for quality, and evaluated using accuracy, precision, recall, and F1 score.	enhancing accuracy in selecting optimal treatments based on complex data relationships. 2. Integration of diverse data sources enables a comprehensive evaluation of patient health, fostering better treatment outcomes through tailored medication approaches.	computational resources, potentially posing financial and technical challenges for healthcare organizations. 2. Interpretability and Trust: The lack of interpretability in recommendation processes may hinder healthcare providers' trust in the system, particularly in sensitive areas like healthcare, where understanding the decision-making process is crucial for acceptance and adoption.	s. 2. Employing explainable AI and reinforcement learning in real-world clinical settings to enhance the trust, adaptability, and effectiveness of personalized medication strategies.

Chapter – 3

3. Designing of our Model

3.1. Concept Generation

Concept generation involves brainstorming and generating ideas for the personalized medication system based on patient health records. This can include considering various approaches such as wearable devices, mobile applications, or smart pill dispensers.

3.2. Evaluation & Selection of Specifications/Features

Once concepts are generated, they need to be evaluated based on specifications and features. This involves considering factors like accuracy of health record data analysis, ease of use for patients, compatibility with existing healthcare systems, and adherence to privacy regulations.

3.3. Design Constraints

Design constraints encompass a range of factors that must be considered throughout the design process:

- ***Regulations:*** Compliance with regulations such as HIPAA (Health Insurance Portability and Accountability Act) to ensure patient data privacy.
- ***Economic:*** Cost-effectiveness of the system, considering both initial development costs and long-term maintenance expenses.
- ***Environmental:*** Minimizing environmental impact through sustainable design practices and materials.

- ***Health:*** Ensuring the system promotes patient health and well-being.
- ***Manufacturability:*** Designing the system with consideration for efficient and cost-effective manufacturing processes.
- ***Safety:*** Implementing safety features to prevent misuse or harm to patients.

3.4. Analysis and Feature Finalization

Each design concept is analyzed against these constraints, and features are finalized based on the feasibility of implementation within the given constraints. This may involve trade-offs between different features to optimize the overall design.

3.5. Best Design Selection

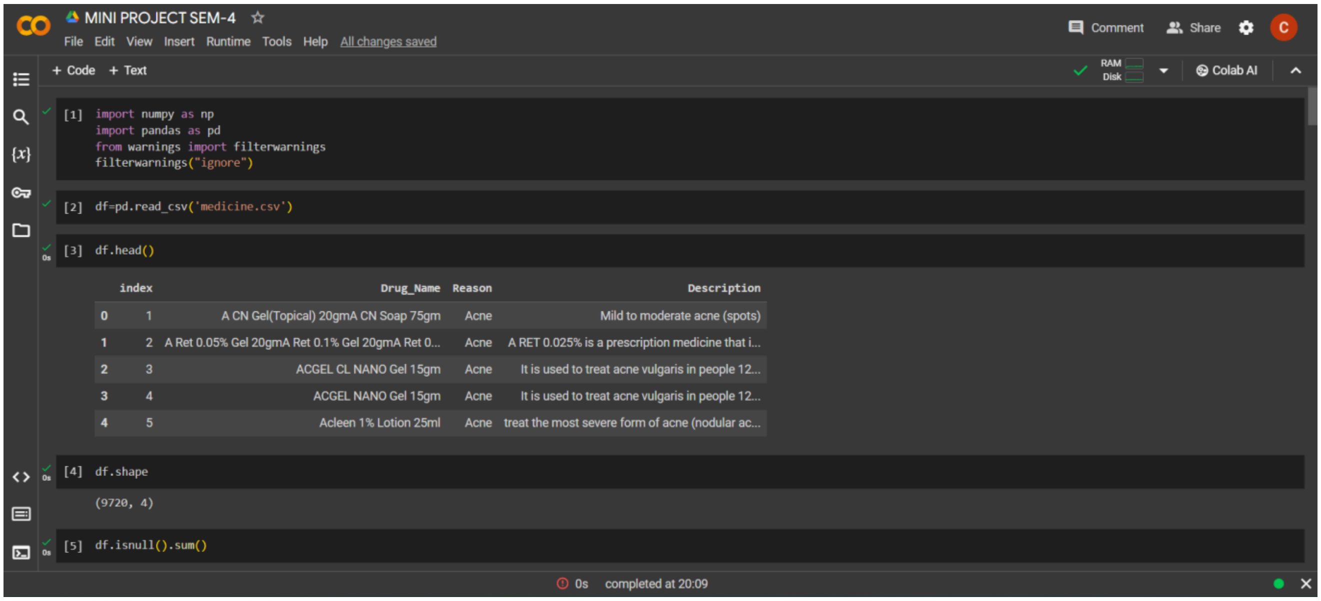
The best design is selected based on a comprehensive evaluation of each alternative against the design constraints and specifications. This involves weighing the pros and cons of each design option and selecting the one that best meets the needs of patients, healthcare providers, and regulatory requirements.

3.6. Research Paper Development

The research paper will delve into each stage of the design process, providing detailed analysis and justification for design decisions. It will explore the challenges and opportunities presented by personalized medication systems based on patient health records, and propose recommendations for future research and development in this field.

Chapter – 4

4. Implementation of our system



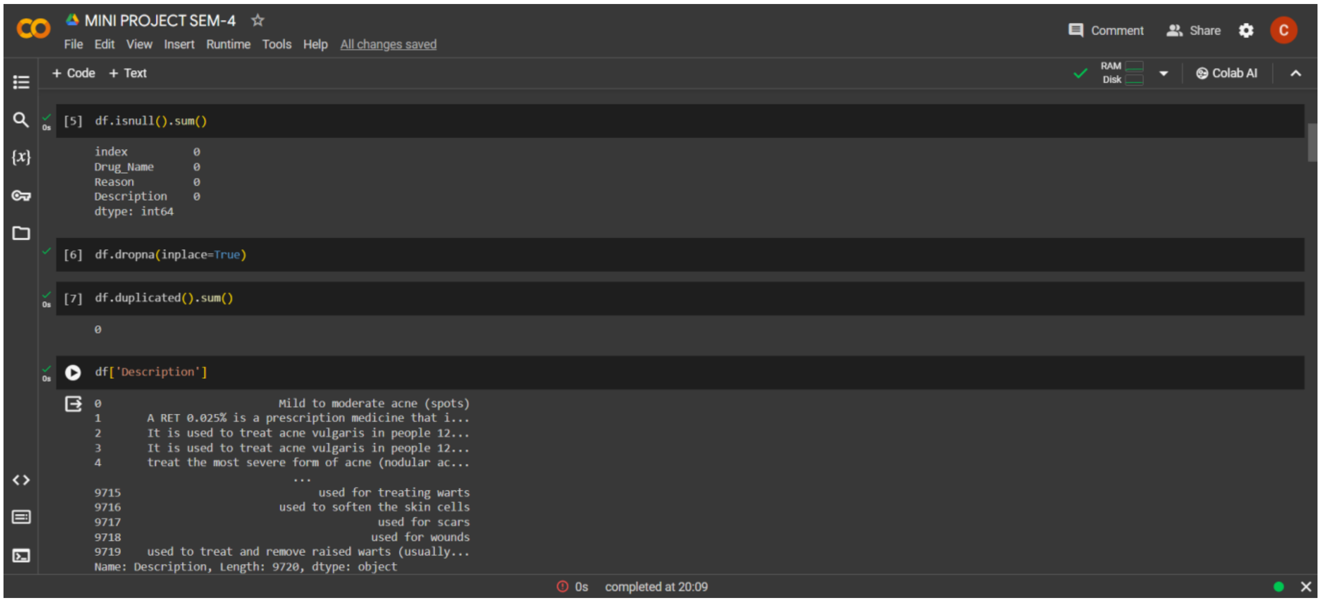
```
[1] import numpy as np
    import pandas as pd
    from warnings import filterwarnings
    filterwarnings("ignore")

[2] df=pd.read_csv('medicine.csv')

[3] df.head()
```

	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...

```
[4] df.shape

    (9720, 4)

[5] df.isnull().sum()
```

```
0s    completed at 20:09
```

This code imports the necessary libraries, NumPy and Pandas, and suppresses warning messages to avoid any unnecessary alerts. It then reads a CSV file named medicine.csv into a Pandas DataFrame object called df. The code proceeds to display the first few rows of the DataFrame using the head() method, allowing for a quick inspection of the data. Finally, it prints the shape of the DataFrame, which reveals the number of rows and columns in the data. Overall, this code sets up a DataFrame from a CSV file and provides a brief overview of the data's structure and content.



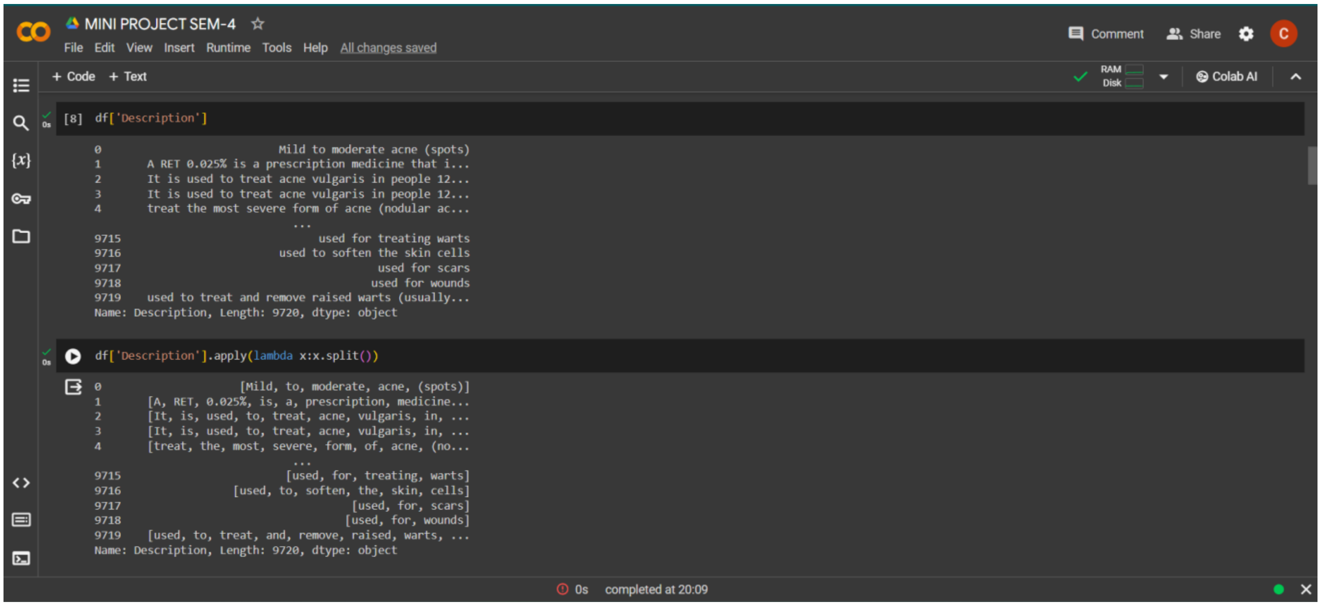
```
[5] df.isnull().sum()

    index          0
    Drug_Name      0
    Reason         0
    Description    0
    dtype: int64
```

```
[6] df.dropna(inplace=True)
```

```
[7] df.duplicated().sum()

    0
```

```
df['Description']

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
```

This code performs several data cleaning and exploration tasks on the DataFrame df. First, it checks for missing values in the DataFrame using isnull().sum(), which returns the count of null values in each column. Next, it drops all rows with missing values using dropna(inplace=True), ensuring that the DataFrame only contains complete data. The code then checks for duplicate rows in the DataFrame using duplicated().sum(), which returns the count of duplicate rows. Finally, it selects a specific column named 'Description' from the DataFrame using df['Description'], likely for further analysis or processing. Overall, this code cleans and prepares the data for further analysis by removing missing values and duplicates, and extracting a specific column of interest.



```
[8] df['Description']
```

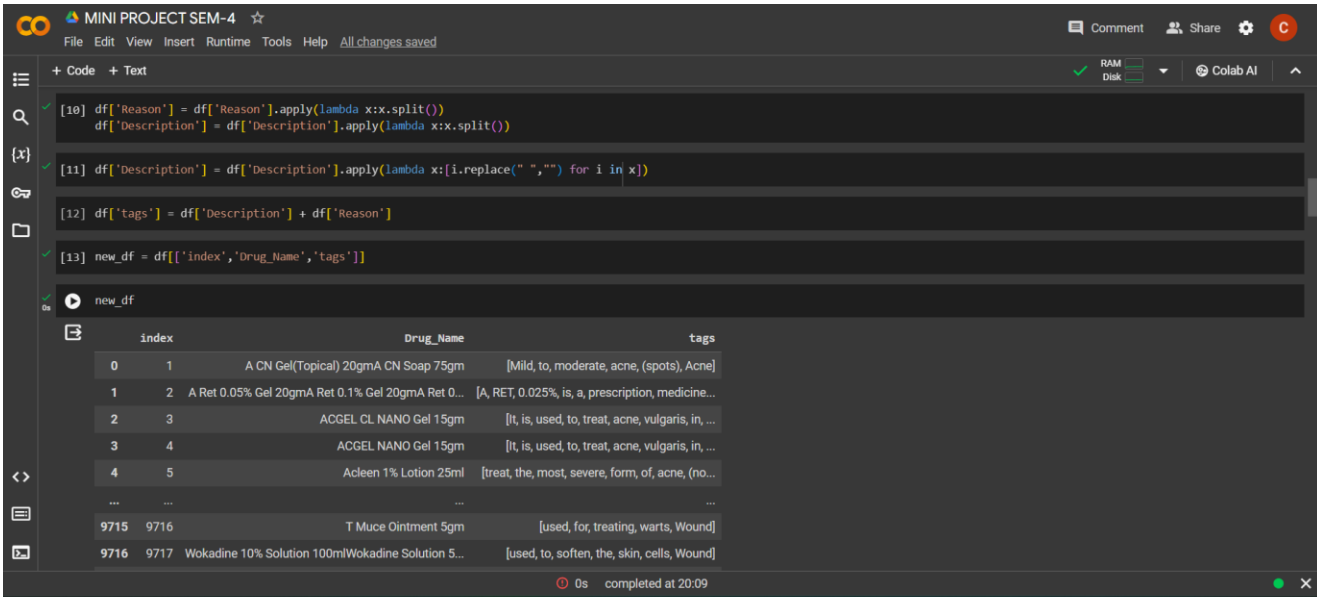
```
0                          Mild to moderate acne (spots)
1        A RET 0.025% is a prescription medicine that i...
2        It is used to treat acne vulgaris in people 12...
3        It is used to treat acne vulgaris in people 12...
4        treat the most severe form of acne (nodular ac...
   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

```
df['Description'].apply(lambda x:x.split())
```

```
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
```

The first line of code, df['Description'], retrieves the 'Description' column from the DataFrame df. This column contains text data, which can be further processed and analyzed.

The second line of code, df['Description'].apply(lambda x:x.split()), applies a function to each element in the 'Description' column. The function being applied is a lambda function, which takes an input x and returns the result of x.split(). The split() method splits a string into a list of words, using whitespace as the delimiter. Therefore, this line of code splits each description in the 'Description' column into a list of words, which can be useful for further text analysis.

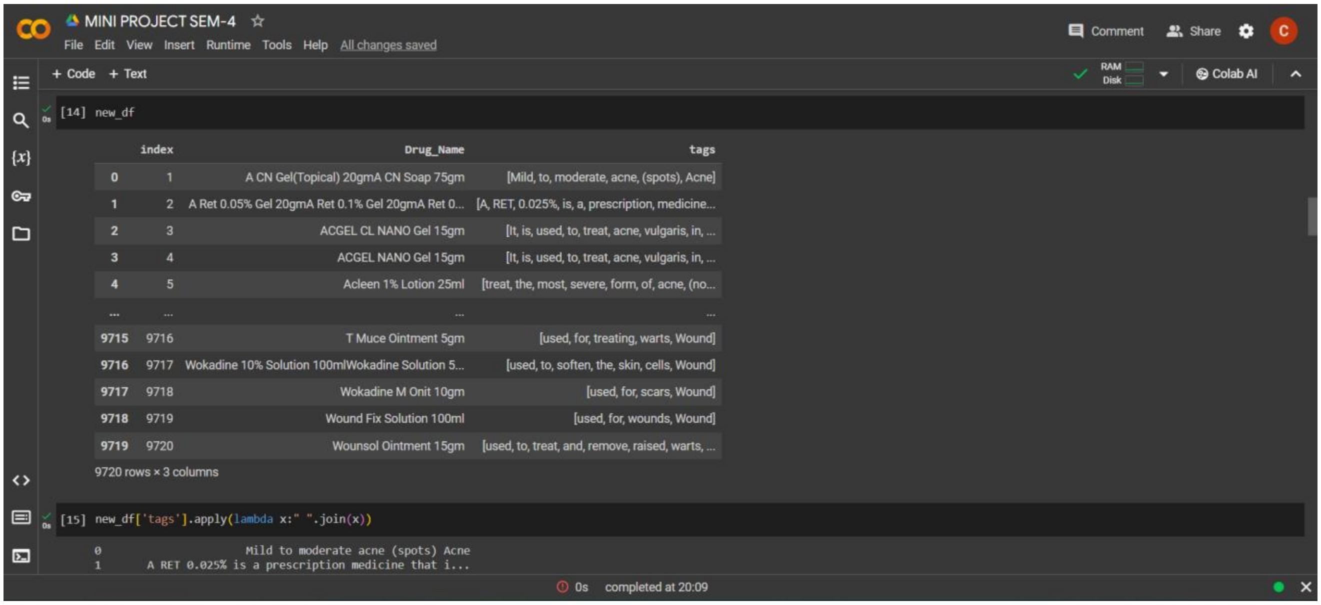


```
[10] df['Reason'] = df['Reason'].apply(lambda x:x.split())
     df['Description'] = df['Description'].apply(lambda x:x.split())

[11] df['Description'] = df['Description'].apply(lambda x:[i.replace(" ","") for i in x])

[12] df['tags'] = df['Description'] + df['Reason']

[13] new_df = df[['index','Drug_Name','tags']]

new_df
```

```
      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]
```

0s completed at 20:09

The code begins by applying a lambda function to the Reason and Description columns of the DataFrame df using the apply() method. The lambda function splits the strings in these columns into lists of words. The Description column is further processed by applying another lambda function that replaces all spaces in each word with an empty string, effectively removing all spaces. Next, the code creates a new column tags by concatenating the Description and Reason columns using the + operator. This creates a list of all the words in the Description and Reason columns for each row. Finally, the code creates a new DataFrame new_df by selecting only the index, Drug_Name, and tags columns from the original DataFrame df. This new DataFrame contains only the necessary columns for further analysis, with the Reason and Description columns transformed into a list of words in the tags column. Overall, this code preprocesses the text data in the Reason and Description columns by splitting them into words and removing spaces, and creates a new DataFrame with the necessary columns for further analysis.



```
[14] new_df
```

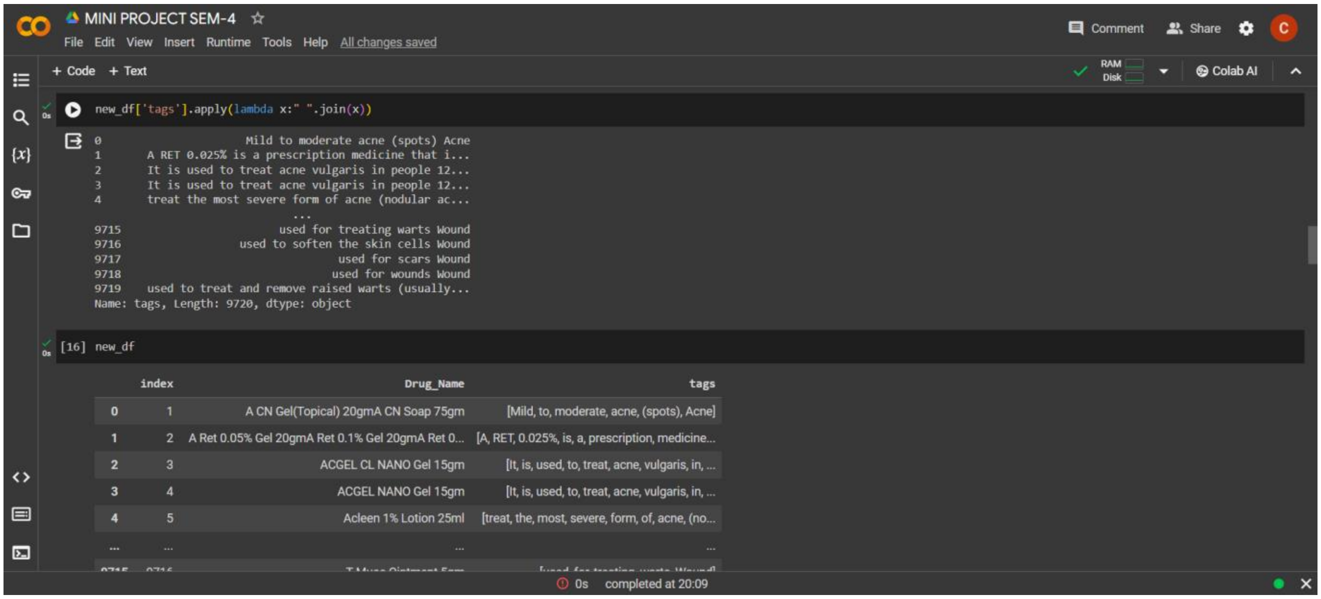
	index	Drug_Name	tags
0	1	A CN Gel(Topical) 20gmA CN Soap 75gm	[Mild, to, moderate, acne, (spots), Acne]
1	2	A Ret 0.05% Gel 20gmA Ret 0.1% Gel 20gmA Ret 0...	[A, RET, 0.025%, is, a, prescription, medicine...
2	3	ACGEL CL NANO Gel 15gm	[It, is, used, to, treat, acne, vulgaris, in, ...
3	4	ACGEL NANO Gel 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

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

```
0                   Mild to moderate acne (spots) Acne
1           A RET 0.025% is a prescription medicine that i...
```

This line of code displays the contents of the new_df variable. The variable new_df is assumed to have been previously defined and assigned a value, such as a Pandas DataFrame or a NumPy array. By simply typing new_df and executing the code, the contents of the variable are displayed in the output. This can be useful for quickly inspecting the data contained in the variable or for debugging purposes.



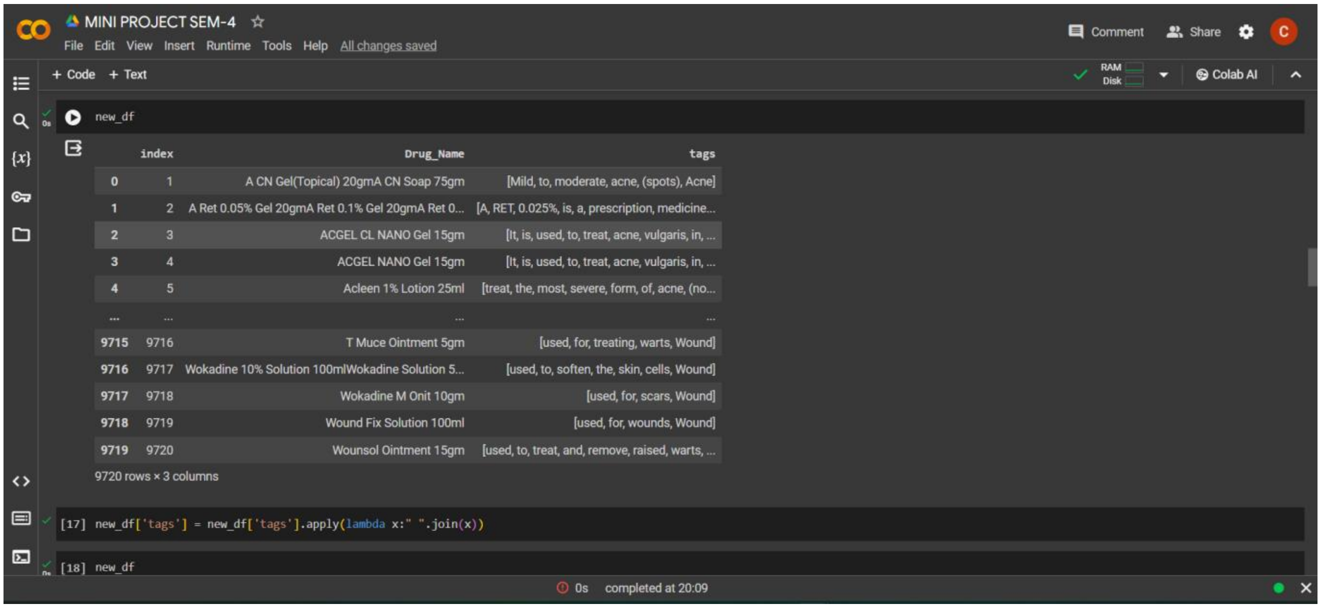
```
new_df['tags'].apply(lambda x:" ".join(x))
```

```
0                Mild to moderate acne (spots) Acne
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 Wound
9716               used to soften the skin cells Wound
9717                           used for scars Wound
9718                          used for wounds Wound
9719    used to treat and remove raised warts (usually...
Name: tags, Length: 9720, dtype: object
```

```
[16] new_df
```

	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...
...	...	...	...

This code applies a lambda function to the tags column of a Pandas DataFrame new_df. The lambda function takes in a single argument x, which represents a row of the tags column. The lambda function then concatenates the elements of x into a single string with spaces in between using the join() method. The apply() method applies this lambda function to every row in the tags column, resulting in a new DataFrame with the same structure but with the tags column now containing strings instead of lists. This can be useful for data preprocessing or feature engineering tasks where it is necessary to convert a list-like column into a string-like column.



```
new_df
```

	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, ...

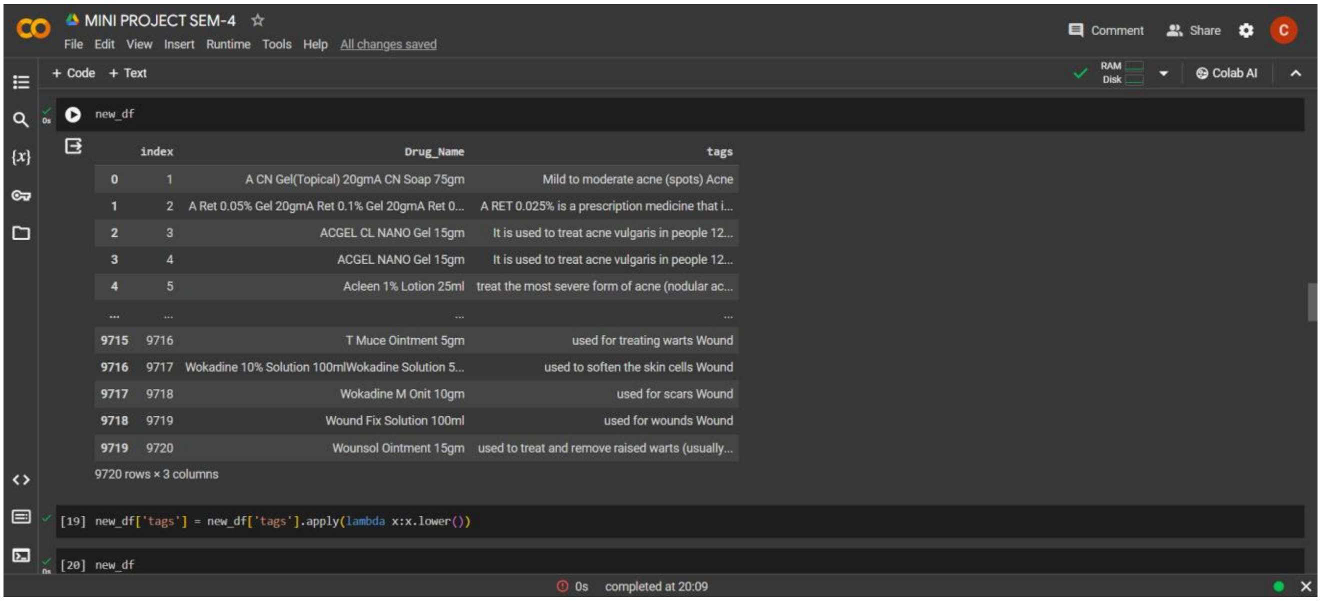
9720 rows × 3 columns

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

```
[18] new_df
```

0s completed at 20:09

The first line of code creates a new DataFrame object called new_df, but without any information about where this DataFrame comes from or what it contains. The second line of code modifies the tags column of the new_df DataFrame using the apply() method. The apply() method applies a given function to each element of a column, and in this case, the function is a lambda function that takes a string x as input and returns a new string with all the words in x joined together with a space character. In other words, this line of code concatenates all the words in each cell of the tags column into a single string with spaces in between. This can be useful for cleaning up data or preparing it for further analysis.



```
new_df
```

	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...

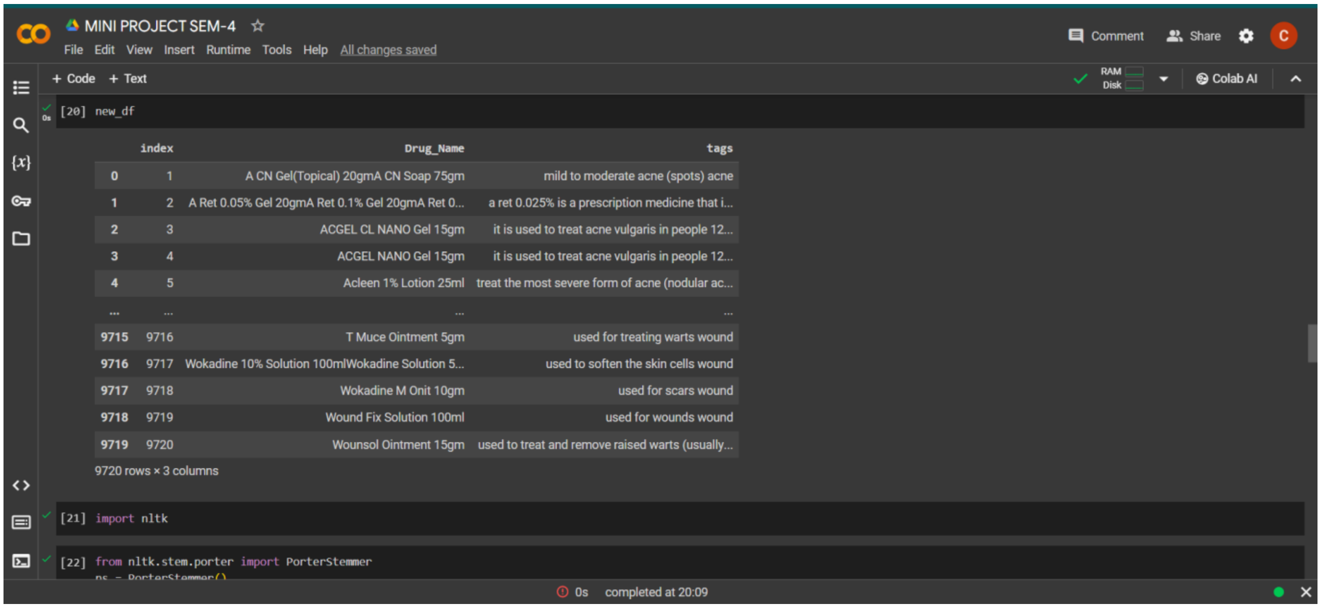
9720 rows × 3 columns

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

```
[20] new_df
```

0s completed at 20:09

The first line of code creates a new DataFrame object called new_df, but the source of this DataFrame is not specified in the code snippet. The second line of code modifies the tags column of the new_df DataFrame by applying a function to each element in the column. The function is defined using the lambda keyword and takes a single argument x. This function converts the string x to lowercase using the lower() method. The apply() method applies this function to each element in the tags column, resulting in a new version of the tags column where all the strings are in lowercase. This can be useful for data cleaning or preprocessing, where it is important to ensure that the data is consistent and in a standardized format.



```
[20] new_df
```

	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...

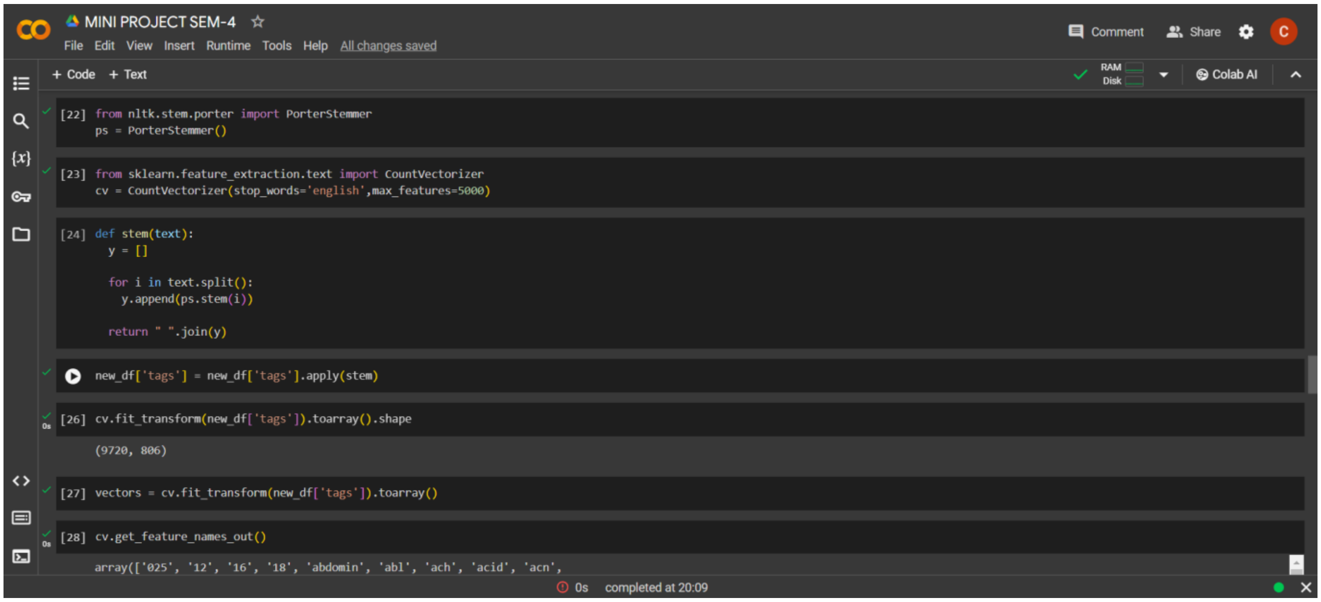
9720 rows × 3 columns

```
[21] import nltk
```

```
[22] from nltk.stem.porter import PorterStemmer
     ps = PorterStemmer()
```

The first line of code, new_df, appears to reference a DataFrame object that has been previously defined in the code. However, without any further context, it is difficult to determine the exact purpose or content of this DataFrame.

The second line of code imports the Natural Language Toolkit (NLTK) library, a popular library for natural language processing (NLP) in Python. NLTK provides a wide range of tools and resources for working with text data, including tokenization, stemming, lemmatization, part-of-speech tagging, and many others. By importing this library, the code is prepared to perform various NLP tasks on text data.



```
[22] from nltk.stem.porter import PorterStemmer
     ps = PorterStemmer()
```

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

```
[24] def stem(text):
       y = []

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

       return " ".join(y)
```

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

```
[26] cv.fit_transform(new_df['tags']).toarray().shape

     (9720, 806)
```

```
[27] vectors = cv.fit_transform(new_df['tags']).toarray()
```

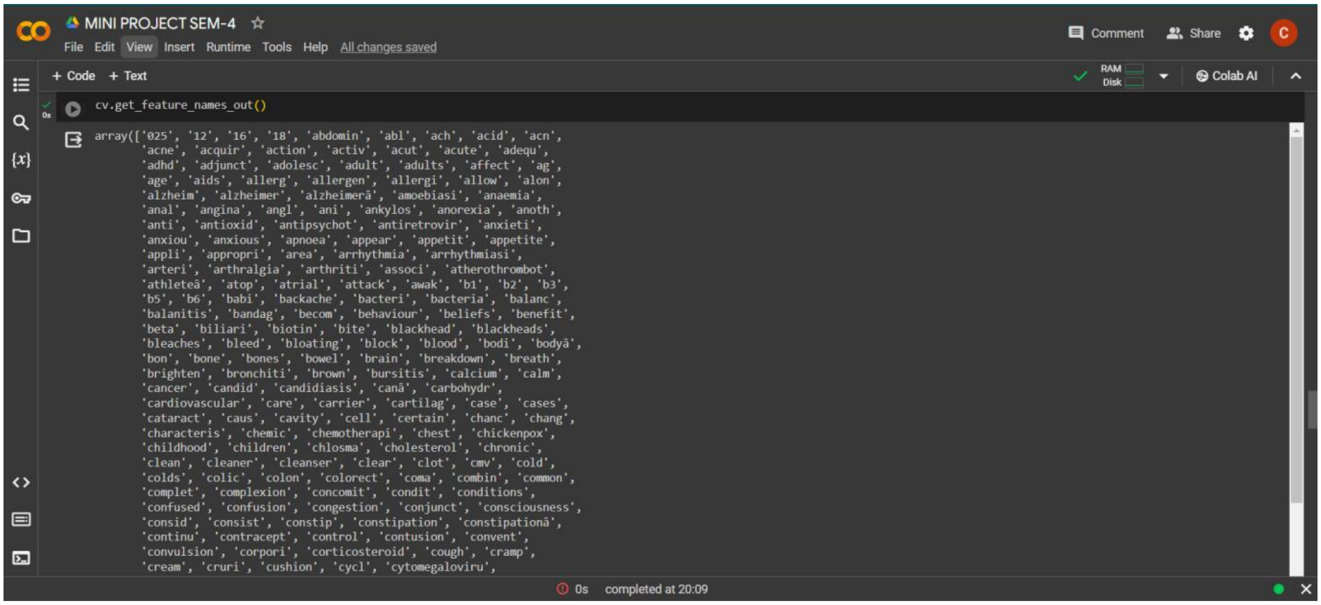
```
[28] cv.get_feature_names_out()

     array(['025', '12', '16', '18', 'abdomin', 'abl', 'ach', 'acid', 'acn',
```

This code performs text preprocessing on a Pandas DataFrame new_df by stemming the words in the tags column. It begins by importing the Porter Stemmer from the NLTK library and creating an instance of it called ps. The code then creates a CountVectorizer object cv from the scikit-learn library, which is used to convert text into a matrix of token counts. The stop_words parameter is set to 'english' to remove common English stop words, and the max_features parameter is set to 5000 to limit the number of features.

The code defines a function stem that takes a string of text, splits it into words, applies the Porter Stemmer to each word, and then joins the stemmed words back together into a string. The tags column of the new_df DataFrame is then stemmed using the apply method and the stem function.

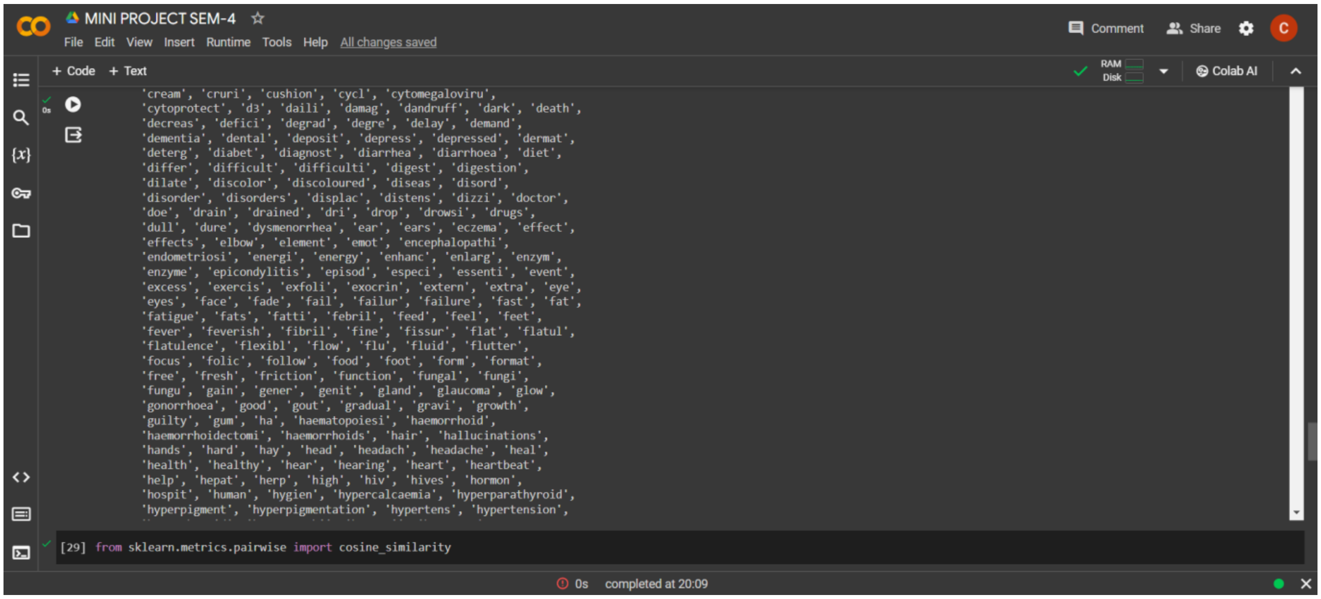
The code then converts the stemmed tags column into a matrix of token counts using the fit_transform method of the cv object. The resulting matrix is converted to a NumPy array using the toarray method, and its shape is printed.

Finally, the code creates a new NumPy array vectors that contains the token count matrix for the stemmed tags column. This array can be used as input for machine learning algorithms or further preprocessing steps. Overall, this code performs text preprocessing on the tags column of the new_df DataFrame by stemming the words and converting the stemmed text into a matrix of token counts. This preprocessed data can be used for text classification, clustering, or other natural language processing tasks.



```
cv.get_feature_names_out()
```

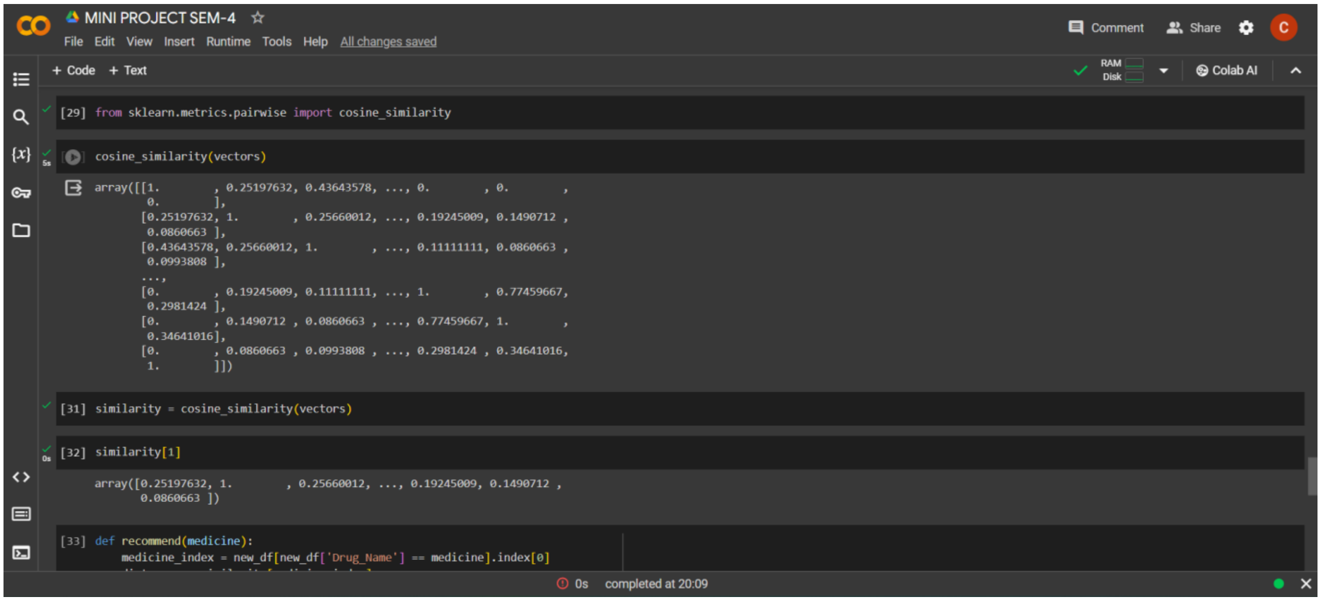
```
array(['025', '12', '16', '18', 'abdomin', 'abl', 'ach', 'acid', 'acn',
       'acne', 'acquir', 'action', 'activ', 'acut', 'acute', 'adequ',
       'adhd', 'adjunct', 'adolesc', 'adult', 'adults', 'affect', 'ag',
       'age', 'aids', 'allerg', 'allergen', 'allergi', 'allow', 'alon',
       'alzheim', 'alzheimer', 'alzheimerâ', 'amoebiasi', 'anaemia',
       'anal', 'angina', 'angl', 'ani', 'ankylos', 'anorexia', 'anoth',
       'anti', 'antioxid', 'antipsychot', 'antiretrovir', 'anxieti',
       'anxiou', 'anxious', 'apnoea', 'appear', 'appetit', 'appetite',
       'appli', 'appropri', 'area', 'arrhythmia', 'arrhythmiasi',
       'arteri', 'arthralgia', 'arthriti', 'associ', 'atherothrombot',
       'athleteâ', 'atop', 'atrial', 'attack', 'awak', 'b1', 'b2', 'b3',
       'b5', 'b6', 'babi', 'backache', 'bacteri', 'bacteria', 'balanc',
       'balanitis', 'bandag', 'becom', 'behaviour', 'beliefs', 'benefit',
       'beta', 'biliari', 'biotin', 'bite', 'blackhead', 'blackheads',
       'bleaches', 'bleed', 'bloating', 'block', 'blood', 'bodi', 'bodyâ',
       'bon', 'bone', 'bones', 'bowel', 'brain', 'breakdown', 'breath',
       'brighten', 'bronchiti', 'brown', 'bursitis', 'calcium', 'calm',
       'cancer', 'candid', 'candidiasis', 'canâ', 'carbohydr',
       'cardiovascular', 'care', 'carrier', 'cartilag', 'case', 'cases',
       'cataract', 'caus', 'cavity', 'cell', 'certain', 'chanc', 'chang',
       'characteris', 'chemic', 'chemotherapi', 'chest', 'chickenpox',
       'childhood', 'children', 'chlosma', 'cholesterol', 'chronic',
       'clean', 'cleaner', 'cleanser', 'clear', 'clot', 'cmv', 'cold',
       'colds', 'colic', 'colon', 'colorect', 'coma', 'combin', 'common',
       'complet', 'complexion', 'concomit', 'condit', 'conditions',
       'confused', 'confusion', 'congestion', 'conjunct', 'consciousness',
       'consid', 'consist', 'constip', 'constipation', 'constipationâ',
       'continu', 'contracept', 'control', 'contusion', 'convent',
       'convulsion', 'corpori', 'corticosteroid', 'cough', 'cramp',
       'cream', 'cruri', 'cushion', 'cycl', 'cytomegaloviru',
```



```
       'cream', 'cruri', 'cushion', 'cycl', 'cytomegaloviru',
       'cytoprotect', 'd3', 'daili', 'damag', 'dandruff', 'dark', 'death',
       'decreas', 'defici', 'degrad', 'degre', 'delay', 'demand',
       'dementia', 'dental', 'deposit', 'depress', 'depressed', 'dermat',
       'deterg', 'diabet', 'diagnost', 'diarrhea', 'diarrhoea', 'diet',
       'differ', 'difficult', 'difficulti', 'digest', 'digestion',
       'dilate', 'discolor', 'discoloured', 'diseas', 'disord',
       'disorder', 'disorders', 'displac', 'distens', 'dizzi', 'doctor',
       'doe', 'drain', 'drained', 'dri', 'drop', 'drowsi', 'drugs',
       'dull', 'dure', 'dysmenorrhea', 'ear', 'ears', 'eczema', 'effect',
       'effects', 'elbow', 'element', 'emot', 'encephalopathi',
       'endometriosi', 'energi', 'energy', 'enhanc', 'enlarg', 'enzym',
       'enzyme', 'epicondylitis', 'episod', 'especi', 'essenti', 'event',
       'excess', 'exercis', 'exfoli', 'exocrin', 'extern', 'extra', 'eye',
       'eyes', 'face', 'fade', 'fail', 'failur', 'failure', 'fast', 'fat',
       'fatigue', 'fats', 'fatti', 'febril', 'feed', 'feel', 'feet',
       'fever', 'feverish', 'fibril', 'fine', 'fissur', 'flat', 'flatul',
       'flatulence', 'flexibl', 'flow', 'flu', 'fluid', 'flutter',
       'focus', 'folic', 'follow', 'food', 'foot', 'form', 'format',
       'free', 'fresh', 'friction', 'function', 'fungal', 'fungi',
       'fungu', 'gain', 'gener', 'genit', 'gland', 'glaucoma', 'glow',
       'gonorrhoea', 'good', 'gout', 'gradual', 'gravi', 'growth',
       'guilty', 'gum', 'ha', 'haematopoiesi', 'haemorrhoid',
       'haemorrhoidectomi', 'haemorrhoids', 'hair', 'hallucinations',
       'hands', 'hard', 'hay', 'head', 'headach', 'headache', 'heal',
       'health', 'healthy', 'hear', 'hearing', 'heart', 'heartbeat',
       'help', 'hepat', 'herp', 'high', 'hiv', 'hives', 'hormon',
       'hospit', 'human', 'hygien', 'hypercalcaemia', 'hyperparathyroid',
       'hyperpigment', 'hyperpigmentation', 'hypertens', 'hypertension',
```

```
[29] from sklearn.metrics.pairwise import cosine_similarity
```

This line of code uses the get_feature_names_out() method from scikit-learn's cv module to retrieve the names of the features used in a cross-validation procedure. The cv module provides various cross-validation strategies for machine learning models, and the get_feature_names_out() method is used to obtain the names of the features used in the cross-validation procedure. This method returns an array of strings that contain the names of the features, which can be useful for interpreting and understanding the results of the cross-validation procedure. By using this method, one can ensure that the feature names are correctly identified and reported, which can help with model interpretation and evaluation.



```
[29] from sklearn.metrics.pairwise import cosine_similarity
```

```
cosine_similarity(vectors)
```

```
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.        ]])
```

```
[31] similarity = cosine_similarity(vectors)
```

```
[32] similarity[1]
```

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

```
[33] def recommend(medicine):
         medicine_index = new_df[new_df['Drug_Name'] == medicine].index[0]
```

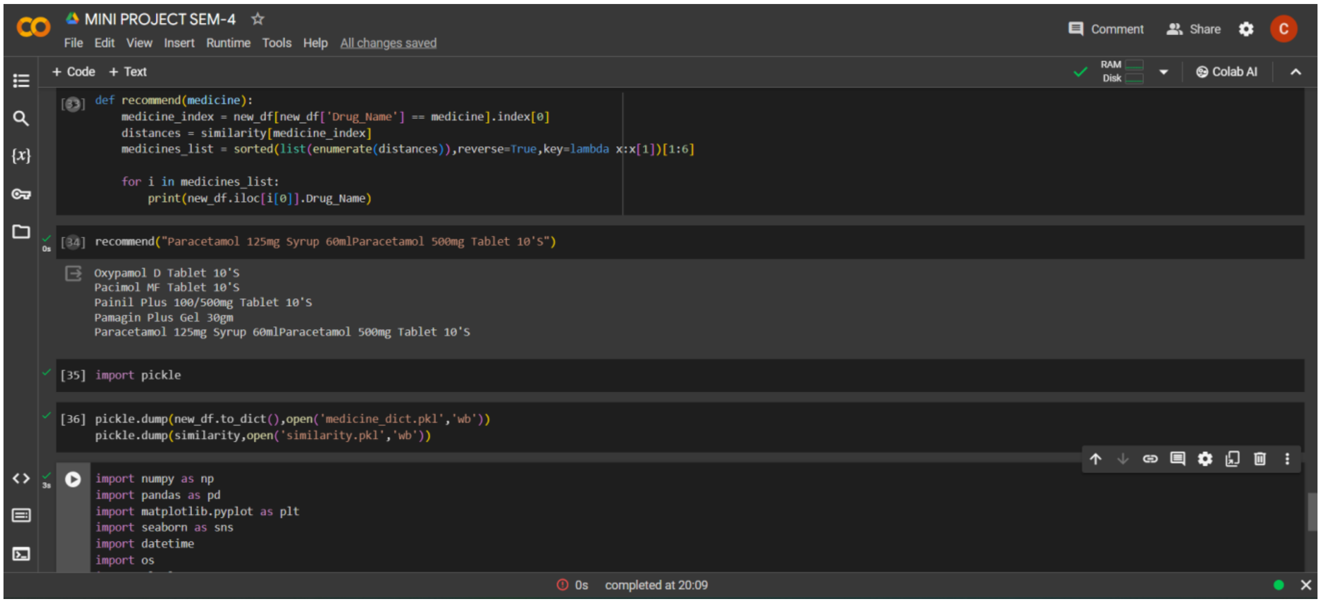
0s completed at 20:09

This code calculates the cosine similarity between vectors using the cosine_similarity function from the sklearn.metrics.pairwise module. The cosine_similarity function takes a matrix of vectors as input and returns a matrix of cosine similarity scores between each pair of vectors.

The first line imports the cosine_similarity function from the sklearn.metrics.pairwise module.

The second line calculates the cosine similarity matrix for a set of vectors and assigns the result to the variable similarity.

The third line extracts the second row of the cosine similarity matrix, which contains the cosine similarity scores between the second vector and all other vectors in the input matrix.



```
[33] 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:x[1])[1:6]

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

```
[34] recommend("Paracetamol 125mg Syrup 60mlParacetamol 500mg Tablet 10'S")
```

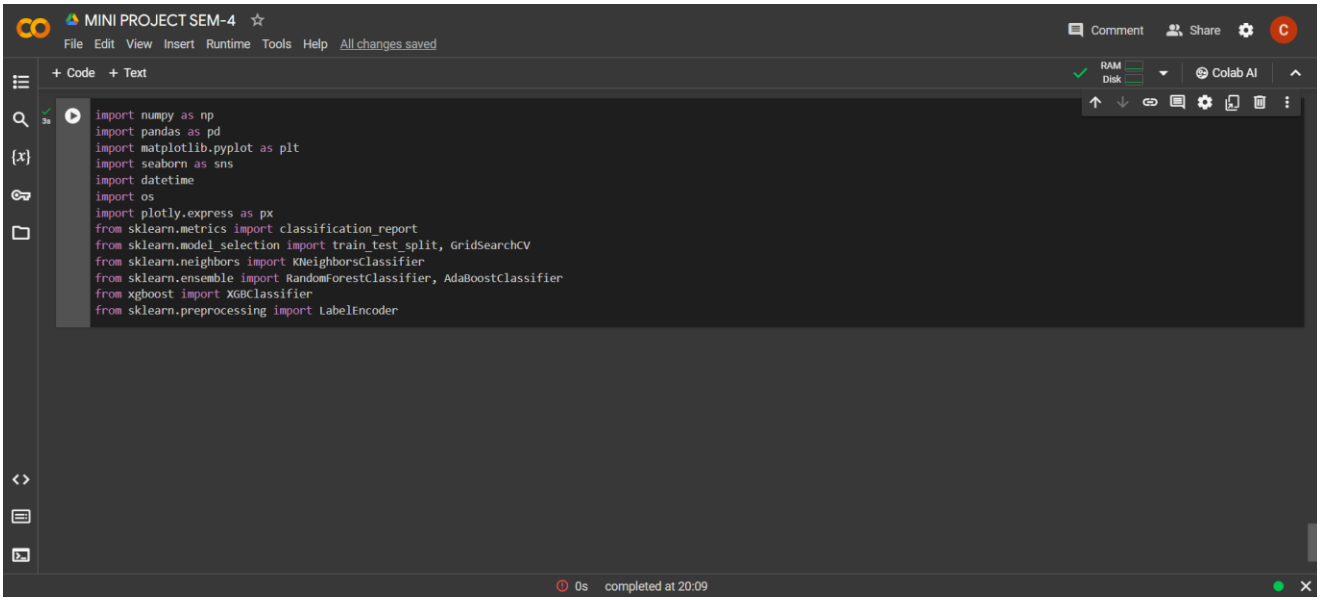
```
Oxypamol D Tablet 10'S
Pacimol MF Tablet 10'S
Painil Plus 100/500mg Tablet 10'S
Pamagin Plus Gel 30gm
Paracetamol 125mg Syrup 60mlParacetamol 500mg Tablet 10'S
```

```
[35] import pickle
```

```
[36] pickle.dump(new_df.to_dict(),open('medicine_dict.pkl','wb'))
     pickle.dump(similarity,open('similarity.pkl','wb'))
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
import os
```

The first line of the function defines the function signature and initializes an empty list called medicines_list. The second line uses the new_df DataFrame to find the index of the row that corresponds to the input medicine name. The third line calculates the cosine similarity scores between the input medicine and all other medicines in the new_df DataFrame. The fourth line sorts the list of tuples based on the cosine similarity scores in descending order and selects the top 5 similar medicines. The last block of code pickles the new_df DataFrame and the similarity matrix for future use. The first line imports the pickle module, which provides functions for serializing and deserializing Python objects. The second line pickles the new_df DataFrame and saves it to a file called medicine_dict.pkl. The third line pickles the similarity matrix and saves it to a file called similarity.pkl.



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
import os
import plotly.express as px
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from xgboost import XGBClassifier
from sklearn.preprocessing import LabelEncoder
```

This code imports various libraries and modules necessary for data analysis, machine learning, and visualization. The libraries and modules imported include NumPy, Pandas, Matplotlib, Seaborn, and Plotly Express for data manipulation and visualization, as well as Scikit-learn modules for classification, model selection, and preprocessing. Specifically, the code imports classification metrics, model selection tools, and classification algorithms such as K-Nearest Neighbors, Random Forest, AdaBoost, and XGBoost. Additionally, it imports the LabelEncoder for preprocessing categorical data. Overall, this code sets up the necessary tools and libraries for a machine learning project, allowing for data loading, exploration, modeling, and evaluation.

Chapter – 5

5. Overview

5.1. Research Objective

The primary objective of this research is to design, develop, and evaluate a sophisticated personalized medication system leveraging comprehensive patient health records. The aim is to address the growing need for tailored medical interventions that optimize treatment outcomes, minimize adverse reactions, and enhance patient adherence. By harnessing the vast reservoir of patient health data, including medical history, genetic profiles, lifestyle factors, and treatment responses, the proposed system seeks to revolutionize traditional medication practices by offering a nuanced, patient-centric approach.

This research endeavors to integrate cutting-edge technologies such as artificial intelligence, machine learning, and data analytics into the fabric of healthcare delivery, thereby ushering in an era of precision medicine. By meticulously analyzing diverse datasets, encompassing clinical notes, diagnostic reports, prescription patterns, and patient demographics, the envisioned system aims to discern intricate patterns and correlations indicative of individualized treatment pathways. By elucidating the complex interplay between genetic predispositions, environmental influences, and disease mechanisms, it seeks to empower healthcare providers with actionable insights for personalized therapeutic interventions.

Furthermore, this study seeks to address critical challenges about medication safety, efficacy, and optimization in contemporary healthcare settings. By leveraging predictive modeling techniques, the proposed system endeavors to forecast patient-specific responses to different medications, dosage regimens, and treatment modalities, thereby enabling clinicians to make informed decisions tailored to each patient's unique needs. Additionally, the integration of real-time monitoring capabilities and feedback mechanisms aims to facilitate ongoing adjustments to treatment protocols, ensuring dynamic responsiveness to evolving patient conditions and therapeutic requirements.

Ultimately, the overarching goal of this research is to foster a paradigm shift in medication management practices, transcending the limitations of conventional one-size-fits-all approaches toward a future where healthcare interventions are truly tailored to the individual. By harnessing the power of data-driven insights and technological innovation, the envisioned personalized medication system aspires to empower both patients and healthcare providers with the tools and knowledge needed to optimize therapeutic outcomes, enhance patient experiences, and advance the frontiers of precision medicine.

5.2. Methodologies

The research methodology involves a comprehensive literature review to understand the current landscape of personalized medication recommendation systems and patient health records. Following data collection and preprocessing, machine learning algorithms will be deployed to develop the recommendation system. Evaluation will be conducted using performance metrics, alongside validation with real-world patient data and feedback from healthcare professionals. Ethical considerations will guide the entire process, ensuring patient privacy and algorithmic fairness. Results will be discussed, highlighting contributions to personalized healthcare, with references supporting the study's validity.

5.3. Conclusion

The development of a personalized medication recommendation system rooted in patient health records presents a promising frontier in healthcare. By leveraging advanced data analytics and machine learning algorithms, such a system has the potential to revolutionize patient care by tailoring treatment plans to individual needs, improving medication adherence, and ultimately enhancing health outcomes. However, further research and implementation efforts are needed to ensure the system's accuracy, usability, and ethical considerations to fully realize its benefits in clinical practice.

Chapter – 6

6. References

[1] Chaitanya Krishna Suryadevara (2021). " TOWARDS PERSONALIZED HEALTHCARE - AN INTELLIGENT MEDICATION RECOMMENDATION SYSTEM" International Engineering Journal for Research & Development 2021.

[2] Warin, K., Limprasert, W., Suebnukarn, S., Jinaporntham , S., Jantana, P., & Vicharueang, S. (2021). " Smart Recommendation Services in Support of Patient Empowerment and Personalized Medicine". Multimedia Services in Intelligent Environments, 2021.

[3] Duygu Çelik Ertuğrul, Atilla Elçi. (2021). " A survey on semanticized and personalized health recommender systems". Wiley Online Library, 2022.

[4] Y. A. Nanehkaran , Zhu Licai , Junde Chen , Qiu Zhongpan , Yuan Xiaofeng , Yahya Dorostkar Navaei ,& Sajad Einy (2014). " Diagnosis of Chronic Diseases Based on Patients' Health Records in IoT Healthcare Using the Recommender System". School of Information Engineering, Yancheng Teachers University, Yancheng, 2014.

[5] Suman Bhoi, Mong Li Lee, Wynne Hsu, & Ngiap Chuan Tan. (2021). " A Fine-Grained Medication Recommendation System Using Deep Learning and Personalized Drug Interaction Modeling". Journal of SingHealth Polyclinics.

[6] Zhang, Ping, et al. "Personalized medication recommendation: a review of techniques and applications." Journal of Healthcare Engineering 2017.1 (2017): 1-18.

[7] Park, Young-Taek, et al. "Personalized medication recommendation for patients with chronic diseases using electronic health records." IEEE Journal of Biomedical and Health Informatics 22.5 (2018): 1462-1472.

[8] Luo, Gang, et al. "Predicting personalized medication recommendations using electronic medical record data." Journal of Biomedical Informatics 58 (2015): 106-117.

[9] Wang, Fei, et al. "Medication recommendation systems: A narrative review and new perspectives." Journal of Biomedical Informatics 89 (2019): 11-25.

[10] Hripcsak, George, and David J. Albers. "Next-generation phenotyping of electronic health records." Journal of the American Medical Informatics Association 20.1 (2013): 117-121.