|  |  |
| --- | --- |
| Project Title | Drugs, Side Effects and Medical Condition arrow\_drop\_up |
| Tools | Python, ML, SQL, Excel |
| Domain | Data Analyst & Data scientist |
| Project Difficulties level | intermediate |

Dataset : Dataset is available in the given link. You can download it at your convenience.

[Click](https://drive.google.com/file/d/1zowlhj1S3NY0zHR0tQMoFv_vpgxdCmYL/view?usp=sharing) [here](https://drive.google.com/file/d/1zowlhj1S3NY0zHR0tQMoFv_vpgxdCmYL/view?usp=sharing) [to](https://drive.google.com/file/d/1zowlhj1S3NY0zHR0tQMoFv_vpgxdCmYL/view?usp=sharing) [download](https://drive.google.com/file/d/1zowlhj1S3NY0zHR0tQMoFv_vpgxdCmYL/view?usp=sharing) [data](https://drive.google.com/file/d/1zowlhj1S3NY0zHR0tQMoFv_vpgxdCmYL/view?usp=sharing) [set](https://drive.google.com/file/d/1zowlhj1S3NY0zHR0tQMoFv_vpgxdCmYL/view?usp=sharing)

About Dataset

Data contains details of various drugs (used for conditions like [Acne](https://www.drugs.com/condition/acne.html), [Cancer](https://www.drugs.com/condition/malignant-disease.html),

[Heart](https://www.drugs.com/condition/heart-disease.html) [Disease](https://www.drugs.com/condition/heart-disease.html), etc. ) and their side effects

Drugs detail URLs were collected from [following](https://www.kaggle.com/datasets/jithinanievarghese/drugs-related-to-common-treatments) [dataset](https://www.kaggle.com/datasets/jithinanievarghese/drugs-related-to-common-treatments) Major Column Descriptors:

generic\_name:

The chemical name of the drug (not brand name) drug\_classes:

The drug belongs to which drug class, i.e a drug class is a set of medications and other compounds that have a similar chemical structure, the same mechanism of action (i.e. binding to the same biological target), a related mode of action, and/or are used to treat the same disease. brand\_names: brand names in which the drugs are being sold or available in the market. activity:

Activity is based on recent site visitor activity relative to other medications in the list. Data was gathered from [https://www.drugs.com](https://www.drugs.com/) rx\_otc:

Rx-to-OTC switch is the transfer of proven prescription drugs to nonprescription, where

OTC (Over-the-counter) = Medication that can be purchased without a medical prescription

Rx = Prescription Needed

Rx/OTC = Prescription or Over-the-counter.

pregnancy\_category:

1. = Adequate and well-controlled studies have failed to demonstrate a risk to the fetus in the first trimester of pregnancy (and there is no evidence of risk in later trimesters).
2. = Animal reproduction studies have failed to demonstrate a risk to the fetus and there are no adequate and well-controlled studies in pregnant women. C = Animal reproduction studies have shown an adverse effect on the fetus and there are no adequate and well-controlled studies in humans, but potential benefits may warrant use in pregnant women despite potential risks. D = There is positive evidence of human fetal risk based on adverse reaction data from investigational or marketing experience or studies in humans, but potential benefits may warrant use in pregnant women despite potential risks. X = Studies in animals or humans have demonstrated fetal abnormalities and/or there is positive evidence of human fetal risk based on adverse reaction data from investigational or marketing experience, and the risks involved in use in pregnant women clearly outweigh potential benefits.

N = FDA has not classified the drug. csa:

Controlled Substances Act (CSA) Schedule

M = The drug has multiple schedules. The schedule may depend on the exact dosage form or strength of the medication.

U = CSA Schedule is unknown.

N = Is not subject to the Controlled Substances Act.

1. = Has a high potential for abuse. Has no currently accepted medical use in treatment in the United States. There is a lack of accepted safety for use under medical supervision.
2. = Has a high potential for abuse. Has a currently accepted medical use in treatment in the United States or a currently accepted medical use with severe restrictions. Abuse may lead to severe psychological or physical dependence. 3 = Has a potential for abuse less than those in schedules 1 and 2. Has a currently accepted medical use in treatment in the United States. Abuse may lead to moderate or low physical dependence or high psychological dependence.
3. = Has a low potential for abuse relative to those in schedule 3. It has a currently accepted medical use in treatment in the United States. Abuse may lead to limited physical dependence or psychological dependence relative to those in schedule 3.
4. = Has a low potential for abuse relative to those in schedule 4. Has a currently accepted medical use in treatment in the United States. Abuse may lead to limited physical dependence or psychological dependence relative to those in schedule 4.

alcohol:

X = Interacts with Alcohol.

rating:

For ratings, users were asked how effective they found the medicine while considering positive/adverse effects and ease of use (1 = not effective, 10 = most effective).

All other columns are self-explanatory.

NOTE :

1. this project is only for your guidance, not exactly the same you have to create. Here I am trying to show the way or idea of what steps you can follow and how your projects look. Some projects are very advanced (because it will be made with the help of flask, nlp, advance ai, advance DL and some advanced things ) which you can not understand .
2. You can make or analyze your project with yourself, with your idea, make it more creative from where we can get some information and understand about our business. make sure what overall things you have created all things you understand very well.

Example: You can get the basic idea how you can create a project from here

|  |
| --- |
| Here's a beginner-friendly guide to performing data analytics on a dataset involving Drugs, Side Effects, and Medical Conditions. The project will involve exploratory data analysis (EDA) using Python with the specified columns.  Project Title:  Exploratory Data Analysis on Drugs, Side Effects, and Medical Conditions 1. Objective  The goal is to analyze the relationships between drugs, their side effects, and the medical conditions they treat, as well as to explore the ratings and reviews associated with these drugs.  2. Dataset Overview  The dataset contains the following columns:   * drug\_name: Name of the drug. * medical\_condition: The condition the drug is used to treat. * side\_effects: Common side effects of the drug. * generic\_name: The generic name of the drug. * drug\_classes: The class of the drug (e.g., antibiotic, antihistamine). * brand\_names: Brand names under which the drug is sold. * activity: The activity of the drug (e.g., active, inactive). * rx\_otc: Indicates if the drug is prescription (Rx) or over-the-counter (OTC). * pregnancy\_category: The drug's pregnancy risk category. * csa: Controlled Substances Act schedule, if applicable. * alcohol: Interactions with alcohol. |

|  |
| --- |
| * related\_drugs: Other drugs related to the primary drug. * medical\_condition\_description: A brief description of the medical   condition.   * rating: Average user rating of the drug. * no\_of\_reviews: Number of user reviews. * drug\_link: URL link to more information about the drug. * medical\_condition\_url: URL link to more information about the medical   condition.   1. Tools Required    * Python: The primary programming language for data analysis.    * Pandas: For data manipulation and analysis.    * Matplotlib/Seaborn: For data visualization.    * Jupyter Notebook: To write and run Python code. 2. Step-by-Step Guide   Step 1: Import Libraries  import pandas as pd import matplotlib.pyplot as plt import seaborn as sns  Step 2: Load the Dataset  # Load your dataset df = pd.read\_csv('path\_to\_your\_dataset.csv') |

|  |
| --- |
| # Display the first few rows of the dataset df.head()  Step 3: Data Cleaning   * Check for missing values:   # Check for missing values df.isnull().sum()   * Handle missing values:   ○ Drop or fill missing values depending on the context.  # Example: Drop rows with missing values df\_cleaned = df.dropna()  # Or fill missing values with a placeholder df\_filled = df.fillna('Unknown')  Step 4: Basic Data Exploration ● Summary statistics:  # Summary statistics |

|  |
| --- |
| df.describe()   * Distribution of Ratings:   # Distribution of drug ratings plt.figure(figsize=(10, 6)) sns.histplot(df['rating'], bins=10, kde=True) plt.title('Distribution of Drug Ratings') plt.xlabel('Rating') plt.ylabel('Frequency') plt.show()  Step 5: Analyzing Relationships ● Top Drugs by Condition:  # Count the most common drugs for each medical condition top\_drugs = df.groupby('medical\_condition')['drug\_name'].value\_counts().nla rgest(10) print(top\_drugs)   * Side Effects Analysis: |

|  |
| --- |
| # Analyzing the most common side effects side\_effects = df['side\_effects'].value\_counts().head(10) print(side\_effects)   * Drug Ratings by Class:   # Boxplot of ratings by drug class plt.figure(figsize=(12, 8)) sns.boxplot(x='drug\_classes', y='rating', data=df) plt.xticks(rotation=90) plt.title('Drug Ratings by Class') plt.show()  Step 6: Conclusion   * Summarize findings:   ○ Identify any trends or patterns in the data.  ○ Discuss how certain drug classes or conditions are associated with specific side effects or ratings.   1. Next Steps    * Advanced Analysis: Perform more sophisticated statistical tests or machine learning techniques.    * Reporting: Create a report or presentation to share the findings. 2. Example Output |

|  |
| --- |
| * Distribution of Drug Ratings:   1. A histogram showing how drug ratings are distributed, with peaks at certain rating values. * Top Drugs for a Condition:   1. A list or bar chart showing the most commonly prescribed drugs for a particular condition. * Side Effects Analysis:   1. A list of the most common side effects reported in the dataset.   By following this guide, a beginner can start exploring and analyzing the dataset effectively. Let me know if you need further assistance! |

Example: You can get the basic idea how you can create a project from here

Sample code with output

|  |
| --- |
| *# Import dataset* import pandas as pd import numpy as np  *# Read the CSV file into a DataFrame*  fpath =  '/kaggle/input/drugs-side-effects-and-medical-condition/drugs\_s ide\_effects\_drugs\_com.csv' data = pd.read\_csv(fpath)  *# Display the columns quantity and names*  print('The dataset has {} rows and {} columns'.format(data.shape[0], data.shape[1])) print("column:") print(data.columns) |

|  |
| --- |
| The dataset has 2931 rows and 17 columns  column:  Index(['drug\_name', 'medical\_condition', 'side\_effects',  'generic\_name',  'drug\_classes', 'brand\_names', 'activity', 'rx\_otc',  'pregnancy\_category', 'csa', 'alcohol', 'related\_drugs',  'medical\_condition\_description', 'rating',  'no\_of\_reviews', 'drug\_link',  'medical\_condition\_url'], dtype='object')  In [2]:  *# Show the main information about dataset* data.info()  <class 'pandas.core.frame.DataFrame'>  RangeIndex: 2931 entries, 0 to 2930 Data columns (total 17 columns): |

|  |  |  |  |
| --- | --- | --- | --- |
| # | Column | Non-Null Count | Dtype |
| --- | ------ | -------------- | ----- |
| 0 | drug\_name | 2931 non-null | object |
| 1 | medical\_condition | 2931 non-null | object |
| 2 | side\_effects | 2807 non-null | object |
| 3 | generic\_name | 2888 non-null | object |
| 4 | drug\_classes | 2849 non-null | object |
| 5 | brand\_names | 1718 non-null | object |
| 6 | activity | 2931 non-null | object |
| 7 | rx\_otc | 2930 non-null | object |
| 8 | pregnancy\_category | 2702 non-null | object |
| 9 | csa | 2931 non-null | object |
| 10 | alcohol | 1377 non-null | object |
| 11 | related\_drugs | 1462 non-null | object |
| 12 | medical\_condition\_description | 2931 non-null | object |
| 13 | rating | 1586 non-null | float64 |
| 14 | no\_of\_reviews | 1586 non-null | float64 |

15

drug\_link

2931

non-null

object

16

medical\_condition\_url

2931

non-null

object

dtypes:

float64(2),

object(15)

memory

usage:

389.4+

KB

In

[3]:

data

.

head()

Out[3]:

dr

u

g

\_

n

a

m

e

m

edi

cal

\_c

on

diti

on

s

i

d

e

\_

e

ff

e

c

t

s

g

e

n

er

ic

\_

n

a

m

e

dr

u

g

\_c

la

ss

es

br

a

n

d

\_

n

a

m

e

s

a

c

t

i

v

i

t

y

r

x

\_

o

t

c

pre

gn

an

cy\_

cat

eg

ory

c

s

a

a

l

c

o

h

o

l

related

\_drugs

medi

cal\_c

onditi

on\_d

escri

ption

r

a

t

i

n

g

n

o

\_

of

\_r

e

vi

e

w

s

drug\_lin

k

medical

\_conditi

on\_url

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | |
|  | 0 | d o xy cy  cli  n e | Ac ne | (  h  i  v e s  ,   1. if fi c u lt b   r   1. a   t  h  i  n g  ,  s  w  e  lli | d o xy cy  cli  n e | Mi sc ell  a n e o us a  nti m  al ar  ial s,  Te  tr  ac yc lin es | A ct ic la te  ,  A d o x a  C  K  ,  A d o x a P a  k,  A d o x a | 8  7  % | R  x | D | N | X | amoxici llin:  https://  www.dr ugs.co m/amo xicillin..  . | Acne Other name s:  Acne Vulga  ris;  Black head s; B... | 6  .  8 | 7  6  0.  0 | https://w ww.drug  s.com/d oxycycli ne.html | https://w ww.drug  s.com/c ondition/ acne.ht ml |
|  | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | |
|  |  |  |  | n g  i  n y o u  r ..  . |  |  | T  T, Al o  d.  .. |  |  |  |  |  |  |  |  |  |  |  |
| 1 | s  pi ro n  ol  a  ct  o n e | Ac ne | h  i  v e s  ;  d  if fi c u  lt y b  r  e | s  pi ro n  ol  a  ct  o n e | Al d os  te ro n e  re ce  pt or a  nt a g o | Al d a  ct  o n  e,  C  ar o S pi r | 8 2  % | R  x | C | N | X | amlodi pine: https://  www.dr ugs.co m/amlo dipine.  h... | Acne Other name s:  Acne Vulga  ris;  Black head s; B... | 7  .  2 | 4 4  9.  0 | https://w ww.drug  s.com/s pironola ctone.ht ml | https://w ww.drug  s.com/c ondition/ acne.ht ml |
|  | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | |
|  |  |  |  | a t  h  i  n g  ;  s  w e  lli  n g o  f  y o u r. .. |  | ni st s,  P  ot  as  si  u  m  -s p.  .. |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | m  in  o cy  cli  n | Ac ne | s k  i  n  r  a | m  in  o cy  cli  n | Te  tr  ac yc lin es | D  y n a  ci n, | 4 8  % | R  x | D | N | N  a  N | amoxici llin:  https://  www.dr ugs.co m/amo | Acne Other name s:  Acne  Vulga | 5  .  7 | 4 8  2.  0 | https://w ww.drug  s.com/m inocyclin  e.html | https://w ww.drug  s.com/c ondition/ acne.ht ml |
|  | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | |
|  |  | e |  | s h  , f  e v e r, s  w  o  ll e n g  l  a n d s  , fl u  -l i  k e s y | e |  | M  in  o  ci n,  M  in ol ir  a,  S ol o d y n, Xi  m  in o,  V.  .. |  |  |  |  |  | xicillin..  . | ris;  Black head s; B... |  |  |  |  |
|  | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | |
|  |  |  |  | m  ..  . |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | A  cc ut a n e | Ac ne | p  r  o b  l  e  m  s  w  it h y o u  r  v  i  s  i  o n o  r | is ot re  ti n  oi  n  (o ra  l) | Mi sc ell  a n e o us a  nti  n e o  pl  as  tic s,  Mi sc ell  a n e | N  a  N | 4 1  % | R  x | X | N | X | doxycy cline: https://  www.dr ugs.co m/doxy cycline.  .. | Acne Other name s:  Acne Vulga  ris;  Black head s; B... | 7  .  9 | 6 2  3.  0 | https://w ww.drug  s.com/a ccutane.  html | https://w ww.drug  s.com/c ondition/ acne.ht ml |
|  | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | |
|  |  |  |  | h e a ri n g  ;  m u s c  l  e o  ..  . |  | o us  u.  .. |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | cli  n d a  m  yc  in | Ac ne | h  i  v e s  ;  d  if fi c | cli  n d a  m  yc  in to pi c | To  pi  ca  l  ac n e a g e | C le  o  ci  n  T,  C  li n d | 3 9  % | R  x | B | N | N  a  N | doxycy cline: https://  www.dr ugs.co m/doxy cycline.  .. | Acne Other name s:  Acne Vulga  ris;  Black head s; B... | 7  .  4 | 1 4  6.  0 | https://w ww.drug  s.com/m tm/clind amycin-t opical.... | https://w ww.drug  s.com/c ondition/ acne.ht ml |
|  | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | |
|  |  |  |  | u lt b  r  e a  t  h i  n g  ;  s  w  e  lli  n g o  f  y o u  r ..  . | al | nt s,  V a  gi n  al  a  nti  -i  nf  ec tiv es | a  ci  n  E  T  Z,  C  li n d a  ci n  P,  C  li n d a  g.  .. |  |  |  |  |  |  |  |  |  |  |  |
| In [4]: | | | | | | | | | | | | | | | | | |

|  |
| --- |
| *# Dropping the 'brand\_names' column and delete from dataset* data.drop(columns=['brand\_names'], inplace=True)  In [5]:  *# Find duplicate rows based on all columns* duplicate\_rows= data[data.duplicated()]  *#Count the duplicated rows* duplicate\_count = duplicate\_rows.shape[0] *# Print the count of duplicate rows* print("Count of Duplicate Rows:", duplicate\_count) print(duplicate\_rows) *# Print the duplicate rows*  Count of Duplicate Rows: 0  Empty DataFrame  Columns: [drug\_name, medical\_condition, side\_effects, generic\_name, drug\_classes, activity, rx\_otc, pregnancy\_category, csa, alcohol, related\_drugs, medical\_condition\_description, rating, no\_of\_reviews, |

|  |
| --- |
| drug\_link, medical\_condition\_url]  Index: []  In [6]:  *# Convert 'rating' and 'no\_of\_reviews' attributes to numeric* data['rating'] = pd.to\_numeric(data['rating'], errors='coerce')  *# data['no\_of\_reviews'] = pd.to\_numeric(data['no\_of\_reviews'], errors='coerce')*  print(data.dtypes.value\_counts())  object 14 float64 2  Name: count, dtype: int64  In [7]:  *# Convert 'activity' to string, remove whitespace and '%' character, then convert to float and divide by 100* |

|  |
| --- |
| data['activity'] = data['activity'].astype(str).str.replace(r'\s+', '', regex=True).str.rstrip('%').astype('float')/100  *# Display the updated 'activity' column* print(data['activity'].head())   1. 0.87 2. 0.82 3. 0.48 4. 0.41 5. 0.39   Name: activity, dtype: float64  In [8]:  *# Print the total number of missing values*  print("There are {} missing values in this dataset".format(data.isnull().sum().sum())) |

|  |
| --- |
| print('Number of instances = %d' % (data.shape[0])) print('Number of attributes = %d' % (data.shape[1])) print('Number of missing values:') for col in data.columns:  print('\t%s: %d' % (col,data[col].isna().sum()))  There are 6192 missing values in this dataset  Number of instances = 2931  Number of attributes = 16 Number of missing values: drug\_name: 0 medical\_condition: 0 side\_effects: 124 generic\_name: 43 drug\_classes: 82 activity: 0 rx\_otc: 1 |

|  |
| --- |
| pregnancy\_category: 229 csa: 0 alcohol: 1554 related\_drugs: 1469 medical\_condition\_description: 0 rating: 1345 no\_of\_reviews: 1345 drug\_link: 0 medical\_condition\_url: 0  In [9]:  *# In the alcohol column we have X and null(NaN) values, because the drug can interact with alcohol or not.*  *# Therefore, let's replace the values of ak=lcohol column with boolean values.*  *# Let X will be 1 of interaction, NaN will be 0.*  data['alcohol']=data['alcohol'].replace(np.NaN,'0') data['alcohol']=data['alcohol'].replace({'X': 1}) |

|  |
| --- |
| In [10]:  *# To avoid missing values let's fill them with some information*  *# In our case we will replace all them*  *# Fill the null values in 'side\_effects' and 'related\_drugs' with no* data["side\_effects"] = data['side\_effects'].fillna('Unknown') data["related\_drugs"] = data['related\_drugs'].fillna('Unknown')  In [11]:  *# Fill the null values with 0 as a base for 'rating' and*  *'no\_of\_reviews' columns*  *# It will show that there are no information about it* data["rating"] = data['rating'].fillna('0') data["no\_of\_reviews"] = data['no\_of\_reviews'].fillna('0')  In [12]:  *# Fill the null values with ?* |

|  |
| --- |
| data['generic\_name']=data['generic\_name'].replace(np.NaN,'Unkno wn')  *# Fill the null values with undefined for 'drug\_classes'*  data['drug\_classes']=data['drug\_classes'].replace(np.NaN,'Unkno wn')  In [13]:  *# For these two columns we already have some category values from dataset's description*  *# So, let's check the categorical values*  *# For Rx\_OTC* data["rx\_otc"].unique()  Out[13]:  array(['Rx', 'Rx/OTC', 'OTC', nan], dtype=object)  In [14]: |

|  |
| --- |
| *# For pregnancy categories* data["pregnancy\_category"].unique()  Out[14]:  array(['D', 'C', 'X', 'B', 'N', nan, 'A'], dtype=object)  In [15]:  *# Fill the null value with Unknown as a basic value* data['rx\_otc']=data['rx\_otc'].replace(np.NaN, 'Unknown')  *# Fill the null value with Unknown as a basic value*  data['pregnancy\_category']=data['pregnancy\_category'].replace(n p.NaN, 'Unknown')  data['no\_of\_reviews'] = pd.to\_numeric(data['no\_of\_reviews'], errors='coerce')  print(data.head()) |

|  |
| --- |
| dfs=data.copy()  drug\_name medical\_condition \   1. doxycycline Acne 2. spironolactone Acne 3. minocycline Acne 4. Accutane Acne 5. clindamycin Acne   side\_effects  generic\_name \   1. (hives, difficult breathing, swelling in your ... doxycycline 2. hives ; difficulty breathing; swelling of your... spironolactone 3. skin rash, fever, swollen glands, flu-like sym... minocycline 4. problems with your vision or hearing; muscle o... |

|  |  |
| --- | --- |
| isotretinoin (oral)  4 hives ; difficult breathing; swelling of your ... clindamycin topical  drug\_classes | activity |
| rx\_otc \  0 Miscellaneous antimalarials, Tetracyclines | 0.87 |
| Rx  1 Aldosterone receptor antagonists, Potassium-sp... | 0.82 |
| Rx  2 Tetracyclines | 0.48 |
| Rx  3 Miscellaneous antineoplastics, Miscellaneous u... | 0.41 |
| Rx  4 Topical acne agents, Vaginal anti-infectives  Rx  pregnancy\_category csa alcohol \   1. D N 1 2. C N 1 | 0.39 |

|  |
| --- |
| 1. D N 0 2. X N 1 3. B N 0   related\_drugs \   1. amoxicillin: https://www.drugs.com/amoxicillin... 2. amlodipine: https://www.drugs.com/amlodipine.h... 3. amoxicillin: https://www.drugs.com/amoxicillin... 4. doxycycline: https://www.drugs.com/doxycycline... 5. doxycycline: https://www.drugs.com/doxycycline...   medical\_condition\_description rating  no\_of\_reviews \   1. Acne Other names: Acne Vulgaris; Blackheads; B... 6.8   760.0   1. Acne Other names: Acne Vulgaris; Blackheads; B... 7.2   449.0   1. Acne Other names: Acne Vulgaris; Blackheads; B... 5.7   482.0 |

|  |  |  |
| --- | --- | --- |
| 3 Acne Other names: Acne Vulgaris; Blackheads; B... | | 7.9 |
| 623.0  4 Acne Other names: Acne Vulgaris; Blackheads; B... | | 7.4 |
| 146.0  drug\_link | | \ |
| 0 | https://www.drugs.com/doxycycline.html |  |
| 1 | https://www.drugs.com/spironolactone.html |  |
| 2 | https://www.drugs.com/minocycline.html |  |
| 3 | https://www.drugs.com/accutane.html |  |
| 4 | https://www.drugs.com/mtm/clindamycin-topical....  medical\_condition\_url |  |
| 0 | https://www.drugs.com/condition/acne.html |  |
| 1 | https://www.drugs.com/condition/acne.html |  |
| 2 | https://www.drugs.com/condition/acne.html |  |
| 3 | https://www.drugs.com/condition/acne.html |  |
| 4 | https://www.drugs.com/condition/acne.html |  |

|  |
| --- |
| In [16]:  *# Let's check is there any missing values left*  print("There are {} missing values in this dataset".format(data.isnull().sum().sum())) print('Number of instances = %d' % (data.shape[0])) print('Number of attributes = %d' % (data.shape[1])) print('Number of missing values:') for col in data.columns:  print('\t%s: %d' % (col,data[col].isna().sum()))  There are 0 missing values in this dataset  Number of instances = 2931  Number of attributes = 16 Number of missing values: drug\_name: 0 medical\_condition: 0 |

|  |
| --- |
| side\_effects: 0 generic\_name: 0 drug\_classes: 0 activity: 0 rx\_otc: 0 pregnancy\_category: 0 csa: 0 alcohol: 0 related\_drugs: 0 medical\_condition\_description: 0 rating: 0 no\_of\_reviews: 0 drug\_link: 0 medical\_condition\_url: 0  In [17]:  data\_version2=data.copy() |

|  |
| --- |
| print(data\_version2.head())  *# Print head of dataset to our check*  drug\_name medical\_condition \   1. doxycycline Acne 2. spironolactone Acne 3. minocycline Acne 4. Accutane Acne 5. clindamycin Acne   side\_effects  generic\_name \   1. (hives, difficult breathing, swelling in your ... doxycycline 2. hives ; difficulty breathing; swelling of your... spironolactone 3. skin rash, fever, swollen glands, flu-like sym... minocycline 4. problems with your vision or hearing; muscle o... |

|  |  |
| --- | --- |
| isotretinoin (oral)  4 hives ; difficult breathing; swelling of your ... clindamycin topical  drug\_classes | activity |
| rx\_otc \  0 Miscellaneous antimalarials, Tetracyclines | 0.87 |
| Rx  1 Aldosterone receptor antagonists, Potassium-sp... | 0.82 |
| Rx  2 Tetracyclines | 0.48 |
| Rx  3 Miscellaneous antineoplastics, Miscellaneous u... | 0.41 |
| Rx  4 Topical acne agents, Vaginal anti-infectives  Rx  pregnancy\_category csa alcohol \   1. D N 1 2. C N 1 | 0.39 |

|  |
| --- |
| 1. D N 0 2. X N 1 3. B N 0   related\_drugs \   1. amoxicillin: https://www.drugs.com/amoxicillin... 2. amlodipine: https://www.drugs.com/amlodipine.h... 3. amoxicillin: https://www.drugs.com/amoxicillin... 4. doxycycline: https://www.drugs.com/doxycycline... 5. doxycycline: https://www.drugs.com/doxycycline...   medical\_condition\_description rating  no\_of\_reviews \   1. Acne Other names: Acne Vulgaris; Blackheads; B... 6.8   760.0   1. Acne Other names: Acne Vulgaris; Blackheads; B... 7.2   449.0   1. Acne Other names: Acne Vulgaris; Blackheads; B... 5.7   482.0 |

|  |  |  |
| --- | --- | --- |
| 3 Acne Other names: Acne Vulgaris; Blackheads; B... | | 7.9 |
| 623.0  4 Acne Other names: Acne Vulgaris; Blackheads; B... | | 7.4 |
| 146.0  drug\_link | | \ |
| 0 | https://www.drugs.com/doxycycline.html |  |
| 1 | https://www.drugs.com/spironolactone.html |  |
| 2 | https://www.drugs.com/minocycline.html |  |
| 3 | https://www.drugs.com/accutane.html |  |
| 4 | https://www.drugs.com/mtm/clindamycin-topical....  medical\_condition\_url |  |
| 0 | https://www.drugs.com/condition/acne.html |  |
| 1 | https://www.drugs.com/condition/acne.html |  |
| 2 | https://www.drugs.com/condition/acne.html |  |
| 3 | https://www.drugs.com/condition/acne.html |  |
| 4 | https://www.drugs.com/condition/acne.html |  |

In

[18]:

*#*

*Save*

*the*

*data*

data\_version2

.

to\_csv(

'drugs\_side\_effects\_drugs\_com\_version2.csv

'

,

index

=

False

)

In

[19]:

*#*

*Read*

*the*

*new*

*version*

*dataset*

data\_ver3

=

pd

.

read\_csv(

'drugs\_side\_effects\_drugs\_com\_version2.cs

v'

)

data\_ver3[

"pregnancy\_category"

]

.

unique()

Out[19]:

array(['D',

'C',

'X',

'B',

'N',

'Unknown',

'A'],

dtype=object)

In

[20]:

data\_ver3[

"csa"

]

.

unique()

|  |
| --- |
| Out[20]:  array(['N', '2', '4', 'U', 'M', '5', '3'], dtype=object)  In [21]: data\_ver3["rx\_otc"].unique()  Out[21]:  array(['Rx', 'Rx/OTC', 'OTC', 'Unknown'], dtype=object)  In [22]: data\_ver3["generic\_name"].unique()  Out[22]:  array(['doxycycline', 'spironolactone', 'minocycline', ...,  'fenfluramine', 'phendimetrazine tartrate',  'setmelanotide'], dtype=object)  In [23]:  data\_ver3["medical\_condition"].unique() |

|  |
| --- |
| Out[23]:  array(['Acne', 'ADHD', 'AIDS/HIV', 'Allergies', "Alzheimer's",  'Angina',  'Anxiety', 'Asthma', 'Bipolar Disorder', 'Bronchitis', 'Cancer',  'Cholesterol', 'Colds & Flu', 'Constipation', 'COPD', 'Covid 19',  'Depression', 'Diabetes (Type 1)', 'Diabetes (Type 2)', 'Diarrhea',  'Eczema', 'Erectile Dysfunction', 'Gastrointestinal',  'GERD (Heartburn)', 'Gout', 'Hair Loss', 'Hayfever', 'Herpes',  'Hypertension', 'Hypothyroidism', 'IBD (Bowel)',  'Incontinence',  'Insomnia', 'Menopause', 'Migraine', 'Osteoarthritis',  'Osteoporosis', 'Pain', 'Pneumonia', 'Psoriasis',  'Rheumatoid Arthritis', 'Schizophrenia', 'Seizures',  'Stroke', |

|  |
| --- |
| 'Swine Flu', 'UTI', 'Weight Loss'], dtype=object)  In [24]:  from sklearn.preprocessing import LabelEncoder label\_encoder = LabelEncoder() data\_ver3["csa"]=label\_encoder.fit\_transform(data\_ver3["csa"])  data\_ver3["rx\_otc"]=label\_encoder.fit\_transform(data\_ver3["rx\_o tc"])  data\_ver3["generic\_name"] = label\_encoder.fit\_transform(data\_ver3["generic\_name"])  data\_ver3["medical\_condition"] = label\_encoder.fit\_transform(data\_ver3["medical\_condition"])  data\_ver3["pregnancy\_category"] = label\_encoder.fit\_transform(data\_ver3["pregnancy\_category"])  data\_ver3["side\_effects"] =  label\_encoder.fit\_transform(data\_ver3["side\_effects"])  In [25]:  data\_ver3["generic\_name"].unique() |

Out[25]:

array([

642

,

,

1270

1034

,

...,

729

,

1157

,

1259])

In

[26]:

data\_ver3[

"rx\_otc"

]

.

unique()

Out[26]:

array([1,

2

,

0

,

3])

In

[27]:

data\_ver3[

"csa"

]

.

unique()

Out[27]:

array([5,

0

,

2

,

6

,

,

4

,

3

1])

In

[28]:

data\_ver3[

"side\_effects"

]

.

unique()

Out[28]:

array([

15

,

1972

,

2697

,

...,

1647

,

416

,

1706])

|  |
| --- |
| In [29]: data\_ver3["medical\_condition"].unique()  Out[29]:  array([ 2, 0, 1, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14,  10, 15, 16,  17, 18, 19, 20, 21, 23, 22, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,  34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46])  In [30]: data\_ver3["pregnancy\_category"].unique()  Out[30]:  array([3, 2, 6, 1, 4, 5, 0])  In [31]:  df=pd.DataFrame(data\_ver3,columns=('generic\_name',  'medical\_condition', 'no\_of\_reviews', 'side\_effects', 'rating',  'csa', 'pregnancy\_category', 'rx\_otc', 'alcohol')) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| df.head(10)  Out[31]:   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | generic\_ name | medical\_co ndition | no\_of\_re views | side\_ef fects | rati  ng | cs a | pregnancy\_c ategory | rx\_ otc | alco hol | | 0 | 642 | 2 | 760.0 | 15 | 6.8 | 5 | 3 | 1 | 1 | | 1 | 1270 | 2 | 449.0 | 1972 | 7.2 | 5 | 2 | 1 | 1 | | 2 | 1034 | 2 | 482.0 | 2697 | 5.7 | 5 | 3 | 1 | 0 | | 3 | 903 | 2 | 623.0 | 2570 | 7.9 | 5 | 6 | 1 | 1 | | 4 | 505 | 2 | 146.0 | 1260 | 7.4 | 5 | 1 | 1 | 0 | | 5 | 1270 | 2 | 8.0 | 1971 | 7.6 | 5 | 2 | 1 | 1 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 6 | 1335 | 2 | 439.0 | 1895 | 7.7 | 5 | 2 | 1 | 0 | | 7 | 903 | 2 | 999.0 | 2577 | 8.0 | 5 | 6 | 1 | 1 | | 8 | 1276 | 2 | 96.0 | 2702 | 8.5 | 5 | 3 | 1 | 1 | | 9 | 162 | 2 | 86.0 | 2405 | 7.9 | 5 | 2 | 1 | 0 |   In [32]:  from sklearn.preprocessing import StandardScaler scaler=StandardScaler() scaler.fit(df) scaled\_data=scaler.transform(df) print(scaled\_data)  [[-0.11111578 -1.43400434 5.10119829 ... 0.28892455  -0.17025661  1.06232778] |

|  |
| --- |
| [ 1.50040103 -1.43400434 2.89586941 ... -0.43301735  -0.17025661  1.06232778]  [ 0.89479917 -1.43400434 3.12987537 ... 0.28892455  -0.17025661 -0.94132905]  ...  [ 1.21043065 1.82918864 -0.28802985 ... -0.43301735  -0.17025661  1.06232778]  [ 1.47217383 1.82918864 -0.28802985 ... 1.73280834  -0.17025661  -0.94132905]  [ 1.47217383 1.82918864 -0.28802985 ... 1.73280834  -0.17025661  -0.94132905]]  In [33]:  df\_std = pd.DataFrame(scaler.fit\_transform(df), |

|  |  |  |
| --- | --- | --- |
| columns=df.columns) print(df\_std)  generic\_name medical\_condition  side\_effects rating \ | | no\_of\_reviews |
| 0 -0.111116 | -1.434004 | 5.101198 |
| -1.678954 0.819930  1 1.500401 | -1.434004 | 2.895869 |
| 0.778579 0.925271  2 0.894799 | -1.434004 | 3.129875 |
| 1.689009 0.530244  3 0.558639 | -1.434004 | 4.129719 |
| 1.529527 1.109617  4 -0.462673 | -1.434004 | 0.747269 |
| -0.115526 0.977941  ... ... | ... | ... |
| ... ...  2926 -0.832193 | 1.829189 | -0.167481 |
| 0.757231 1.004277  2927 0.112136 | 1.829189 | -0.288030 |

|  |
| --- |
| 0.370455 -0.970861   1. 1.210431 1.829189 -0.288030   -1.029724 -0.970861   1. 1.472174 1.829189 -0.288030   -1.175392 -0.970861   1. 1.472174 1.829189 -0.288030   0.444545 -0.970861  csa pregnancy\_category rx\_otc alcohol   1. 0.274178 0.288925 -0.170257 1.062328 2. 0.274178 -0.433017 -0.170257 1.062328 3. 0.274178 0.288925 -0.170257 -0.941329 4. 0.274178 2.454750 -0.170257 1.062328 5. 0.274178 -1.154959 -0.170257 -0.941329   ... ... ... ... ...   1. -3.424857 2.454750 -0.170257 1.062328 2. -2.500098 -0.433017 -0.170257 1.062328 3. -3.424857 -0.433017 -0.170257 1.062328 |

|  |
| --- |
| 2929 0.274178 1.732808 -0.170257 -0.941329 2930 0.274178 1.732808 -0.170257 -0.941329  [2931 rows x 9 columns]  In [34]: import seaborn as sns import matplotlib.pyplot as plt  plt.figure(figsize=(12, 8)) sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f") plt.title('Correlation Heatmap') plt.show() |

In

[35]:

*#*

*Read*

*the*

*new*

*version*

*dataset*

data\_ver4

=

pd

.

read\_csv(

'drugs\_side\_effects\_drugs\_com\_version2.csv'

)

*#*

*Importing*

*necessary*

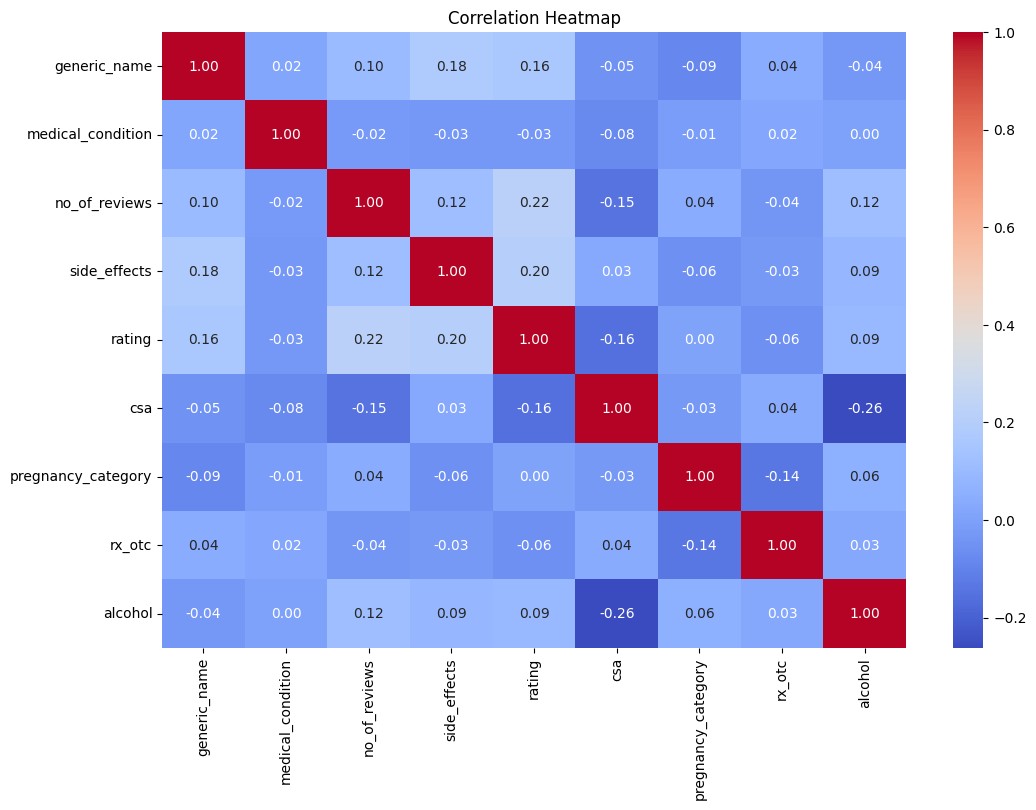
*libraries*

from

mlxtend.frequent\_patterns

import

apriori,



|  |
| --- |
| association\_rules import matplotlib.pyplot as plt import pandas as pd  *# Check for occurrence and frequency of medical conditions, sorted from highest to lowest*  medical\_condition\_counts = data\_ver4['medical\_condition'].value\_counts().sort\_values(ascen ding=False)  print("\nMedical condition occurrence and frequency (sorted from highest to lowest):") print(medical\_condition\_counts)  Medical condition occurrence and frequency (sorted from highest to lowest):  medical\_condition  Pain 264  Colds & Flu 245  Acne 238 |

|  |  |
| --- | --- |
| Hypertension | 177 |
| Osteoarthritis | 129 |
| Hayfever | 124 |
| Eczema | 122 |
| AIDS/HIV | 109 |
| Diabetes (Type 2) | 104 |
| Psoriasis | 93 |
| GERD (Heartburn) | 77 |
| Pneumonia | 72 |
| Angina | 71 |
| Bronchitis | 71 |
| Migraine | 61 |
| Insomnia | 60 |
| Constipation | 60 |
| Diabetes (Type 1) | 57 |
| Osteoporosis | 56 |
| ADHD | 55 |

|  |  |
| --- | --- |
| Depression | 51 |
| Seizures | 50 |
| Bipolar Disorder | 47 |
| UTI | 46 |
| Asthma | 45 |
| Anxiety | 45 |
| Cholesterol | 45 |
| Diarrhea | 38 |
| Covid 19 | 34 |
| Rheumatoid Arthritis | 33 |
| Alzheimer's | 27 |
| Weight Loss | 23 |
| COPD | 23 |
| IBD (Bowel) | 22 |
| Schizophrenia | 20 |
| Cancer | 20 |
| Incontinence | 19 |

|  |
| --- |
| Hypothyroidism 17  Allergies 14  Erectile Dysfunction 13  Hair Loss 11  Herpes 10  Gout 9  Menopause 7  Gastrointestinal 7  Stroke 5  Swine Flu 5  Name: count, dtype: int64  In [36]:  *# Save the results to CSV files if needed*  medical\_condition\_counts.to\_csv('medical\_condition\_counts.csv')  In [37]: |

|  |
| --- |
| *# Importing necessary libraries for processing text* from collections import Counter import re  *# Function to extract side effects from text, split by semicolons* def extract\_side\_effects(text):  *# Split the text on semicolons then strip whitespace*  return [effect.strip() for effect in re.split(r'[;]',  text)]  *# Extract and count occurrences of side effects*  side\_effects = data\_ver4['side\_effects'].dropna().apply(extract\_side\_effects). explode()  side\_effect\_counts = side\_effects.value\_counts().sort\_values(ascending=False)  print("\nSide effects occurrence and frequency (sorted from |

|  |
| --- |
| highest to lowest):") print(side\_effect\_counts)  Side effects occurrence and frequency (sorted from highest to lowest):  side\_effects  hives 1788  difficult breathing  1130  difficulty breathing  450  itching 275  a light-headed feeling, like you might pass out  272  ...  swelling of your face, lips, tongue, or throat. Rizatriptan may cause serious side effects. Stop using rizatriptan and call |

|  |
| --- |
| your doctor at once if you have: sudden and severe stomach pain and bloody diarrhea  1  swelling of your face, lips, tongue, or throat. Report any new or worsening symptoms to your doctor, such as: mood or behavior changes, anxiety , panic attacks , trouble sleeping, or if you feel impulsive, irritable, agitated, hostile, aggressive, restless, hyperactive (mentally or physically), depressed, or have thoughts about suicide or hurting yourself. Zarontin may cause serious side effects. Call your doctor at once if you have: fever, chills, flu symptoms, sore throat , feeling very weak 1  or signs of a stroke--sudden numbness or weakness (especially on one side of the body), sudden severe headache, slurred speech, problems with vision or balance. Common side effects of rizatriptan may include: dizziness , drowsiness, feeling tired  1  Suddenly stopping or reducing the dose of Diastat AcuDial very quickly may precipitate acute withdrawal reactions, which can be life-threatening. In some cases, patients have developed withdrawal symptoms lasting weeks to more than 12 months, including but not limited to: anxiety difficulty thinking mental changes depression insomnia abnormal skin sensations |

|  |
| --- |
| muscle weakness tremors twitching ringing in your ears burning or prickling feeling in your hands, arms, or feet The most frequent side effect reported for Diastat AcuDial in clinical studies was somnolence (sleepiness or drowsiness). Other side effects included dizziness, headache, pain, abdominal pain, nervousness, vasodilation (increase in diameter of blood vessel), diarrhea, ataxia/incoordination (lack of coordination), euphoria (feeling of great happiness or well-being), asthma, rhinitis (irritation of the nose similar to an allergy or a cold), and rash. You are encouraged to report negative side effects of prescription drugs to the FDA. Visit www.fda.gov/medwatch, or call 1-800-FDA-1088. You may also contact Bausch Health Customer Service at 1-800-321-4576. Diastat AcuDial side effects 1  or nausea , vomiting , diarrhea , or stomach pain. 1  Name: count, Length: 8438, dtype: int64  In [38]:  *# Save the side effect counts to a CSV file*  side\_effect\_counts.to\_csv('side\_effect\_counts.csv') |

|  |
| --- |
| In [39]:  *# Function to extract drug classes from text, split by commas* def extract\_drug\_classes(text):  *# Split the text on commas then strip whitespace*  return [effect.strip() for effect in re.split(r'[,]',  text)]  *# Extract and count occurrences of drug classes*  drug\_classes = data\_ver4['drug\_classes'].dropna().apply(extract\_drug\_classes). explode()  drug\_classes\_counts = drug\_classes.value\_counts().sort\_values(ascending=False)  print("\nDrug Classes occurrence and frequency (sorted from highest to lowest):") print(drug\_classes\_counts)  Drug Classes occurrence and frequency (sorted from highest to |

|  |
| --- |
| lowest):  drug\_classes  Upper respiratory combinations 245  Topical acne agents 125  Topical steroids 94  Antihistamines 82  Unknown 82  ...  Immune globulins 1  Smoking cessation agents 1  Mouth and throat products 1  Skeletal muscle relaxant combinations 1  Anthelmintics 1  Name: count, Length: 244, dtype: int64  In [40]:  *# Save the drug classes counts to a CSV file* |

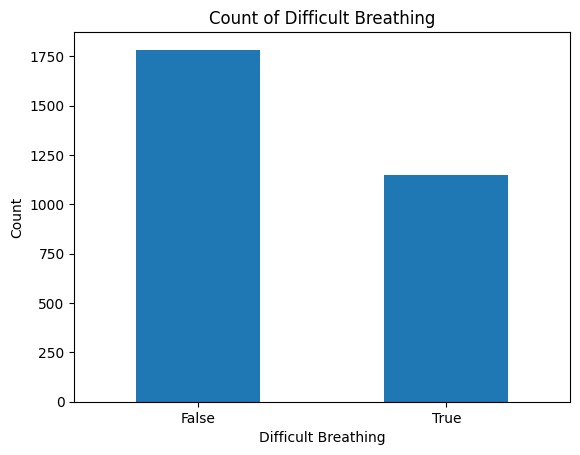
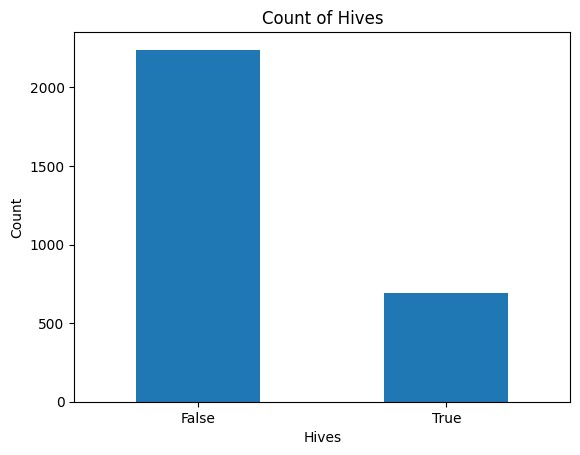
|  |
| --- |
| drug\_classes\_counts.to\_csv('drug\_classes\_counts.csv')  In [41]:  *# Define functions to check for specific side effects and create new boolean columns* def has\_hives(text):  return 'hives' in text.lower() data\_ver4['Hives'] = data\_ver4['side\_effects'].apply(has\_hives)  def has\_difficult\_breathing(text):  return 'difficult breathing' in text.lower() or 'difficulty  breathing' in text.lower()  data\_ver4['Difficult Breathing'] = data\_ver4['side\_effects'].apply(has\_difficult\_breathing)  def has\_itching(text): return 'itching' in text.lower() data\_ver4['Itching'] = |

|  |
| --- |
| data\_ver4['side\_effects'].apply(has\_itching)  In [42]:  *# Define functions to check for specific drug classes and create new boolean columns* def is\_usc(text):  return 'Upper respiratory combinations' in text  data\_ver4['Upper respiratory combinations'] = data\_ver4['drug\_classes'].apply(is\_usc)  def is\_steriods(text):  return 'Topical steroids' in text  data\_ver4['Topical steroids'] = data\_ver4['drug\_classes'].apply(is\_steriods)  def is\_acne(text):  return 'Topical acne agents' in text  data\_ver4['Topical acne agents'] = data\_ver4['drug\_classes'].apply(is\_acne) |

|  |
| --- |
| In [43]:  *# Define functions to check for specific medical conditions and create new boolean columns* def has\_pain(text): return 'Pain' in text  data\_ver4['Pain'] = data\_ver4['medical\_condition'].apply(has\_pain)  def has\_colds\_and\_flu(text): return 'Colds & Flu' in text  data\_ver4['Colds & Flu'] = data\_ver4['medical\_condition'].apply(has\_colds\_and\_flu)  def has\_acne(text): return 'Acne' in text  data\_ver4['Acne'] = data\_ver4['medical\_condition'].apply(has\_acne) |

|  |
| --- |
| In [44]:  *# Plot the count of occurrences for each side effect* import seaborn as sns  *# Plot count of Hives*  data\_ver4['Hives'].value\_counts().plot(kind='bar') plt.title('Count of Hives') plt.xlabel('Hives') plt.ylabel('Count') plt.xticks([0, 1], ['False', 'True'], rotation=0) plt.show()  *# Plot count of Difficult Breathing*  data\_ver4['Difficult  Breathing'].value\_counts().plot(kind='bar') plt.title('Count of Difficult Breathing') plt.xlabel('Difficult Breathing') |

|  |
| --- |
| plt.ylabel('Count') plt.xticks([0, 1], ['False', 'True'], rotation=0) plt.show()  *# Plot count of Itching* data\_ver4['Itching'].value\_counts().plot(kind='bar') plt.title('Count of Itching') plt.xlabel('Itching') plt.ylabel('Count') plt.xticks([0, 1], ['False', 'True'], rotation=0) plt.show() |



In

[45]:

*#*

*Plot*

*the*

*count*

*of*

*occurrences*

*for*

*each*

*drug*

*class*

*#*

*Plot*

*count*

*of*

*Upper*

*respiratory*

*combinations*

data\_ver4[

'Upper

respiratory

combinations'

]

.

value\_counts()

.

plot(kind

=

'bar'

)

plt

.

title(

'Count

of

Upper

respiratory

combinations'

)

plt

.

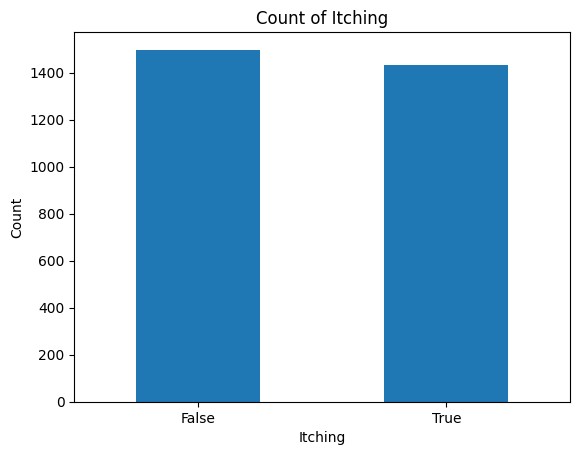
xlabel(

'Upper

respiratory

combinations'

)



|  |
| --- |
| plt.ylabel('Count') plt.xticks([0, 1], ['False', 'True'], rotation=0) plt.show()  *# Plot count of Topical steroids* data\_ver4['Topical steroids'].value\_counts().plot(kind='bar') plt.title('Count of Topical steroids') plt.xlabel('Topical steroids') plt.ylabel('Count') plt.xticks([0, 1], ['False', 'True'], rotation=0) plt.show() |

[Reference](https://github.com/dhyutin/Drugs-Side-Effects-and-Medical-Conditions-Analysis) [link](https://github.com/dhyutin/Drugs-Side-Effects-and-Medical-Conditions-Analysis)