

250k Medicines Usage, Side Effects, and Substitutes

1. Objective

The project aims to explore the relationships between medicines, their substitutes, side effects, and uses. You'll also analyze how these factors vary across different chemical, therapeutic, and action classes.

2. Data Overview

- Basic Info: id, name
- Substitutes: substitute0, substitute1, substitute2, substitute3, substitute4
- Side Effects: sideEffect0 to sideEffect41
- Uses: use0 to use4
- Classification: Chemical Class, Habit Forming, Therapeutic Class, Action Class

3. Data Loading and Initial Exploration

```
In [75]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import missingno as msno
```

```
In [76]: df = pd.read_csv(r"C:\Users\user\OneDrive\Desktop\Data internship\Datasets\Project\250k Medicines Usage, Side Effects and Substitutes.csv")
df.head()
```

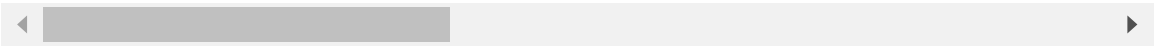
C:\Users\user\AppData\Local\Temp\ipykernel_16432\1496949702.py:1: DtypeWarning: Columns (42,43,44,45,46,47,48) have mixed types. Specify dtype option on import or set low_memory=False.

```
df = pd.read_csv(r"C:\Users\user\OneDrive\Desktop\Data internship\Datasets\Project\250k Medicines Usage, Side Effects and Substitutes.csv")
```

Out[76]:

	id	name	substitute0	substitute1	substitute2	substitute3	substitute4	sideEff
0	1	augmentin 625 duo tablet	Penciclav 500 mg/125 mg Tablet	Moxikind-CV 625 Tablet	Moxiforce-CV 625 Tablet	Fightox 625 Tablet	Novamox CV 625mg Tablet	Vom
1	2	azithral 500 tablet	Zithrocare 500mg Tablet	Azax 500 Tablet	Zady 500 Tablet	Cazithro 500mg Tablet	Trulimax 500mg Tablet	Vom
2	3	ascoril ls syrup	Solvin LS Syrup	Ambrodil-LX Syrup	Zerotuss XP Syrup	Capex LS Syrup	Broxum LS Syrup	Na
3	4	allegra 120mg tablet	Lcfex Tablet	Etofex 120mg Tablet	Nexofex 120mg Tablet	Fexise 120mg Tablet	Histafree 120 Tablet	Head
4	5	avil 25 tablet	Eralet 25mg Tablet	NaN	NaN	NaN	NaN	Sleepi

5 rows × 58 columns



In [77]: df.shape

Out[77]: (248218, 58)

In [78]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 248218 entries, 0 to 248217
Data columns (total 58 columns):
```

#	Column	Non-Null Count	Dtype
0	id	248218 non-null	int64
1	name	248218 non-null	object
2	substitute0	238621 non-null	object
3	substitute1	233867 non-null	object
4	substitute2	230233 non-null	object
5	substitute3	226856 non-null	object
6	substitute4	223962 non-null	object
7	sideEffect0	248218 non-null	object
8	sideEffect1	238416 non-null	object
9	sideEffect2	229500 non-null	object
10	sideEffect3	207638 non-null	object
11	sideEffect4	163560 non-null	object
12	sideEffect5	131258 non-null	object
13	sideEffect6	91857 non-null	object
14	sideEffect7	67750 non-null	object
15	sideEffect8	48506 non-null	object
16	sideEffect9	37708 non-null	object
17	sideEffect10	27274 non-null	object
18	sideEffect11	20331 non-null	object
19	sideEffect12	16282 non-null	object
20	sideEffect13	14727 non-null	object
21	sideEffect14	10419 non-null	object
22	sideEffect15	7681 non-null	object
23	sideEffect16	6009 non-null	object
24	sideEffect17	5382 non-null	object
25	sideEffect18	4515 non-null	object
26	sideEffect19	3946 non-null	object
27	sideEffect20	3223 non-null	object
28	sideEffect21	3125 non-null	object
29	sideEffect22	3048 non-null	object
30	sideEffect23	2905 non-null	object
31	sideEffect24	2723 non-null	object
32	sideEffect25	1503 non-null	object
33	sideEffect26	1503 non-null	object
34	sideEffect27	1494 non-null	object
35	sideEffect28	1494 non-null	object
36	sideEffect29	1438 non-null	object
37	sideEffect30	1329 non-null	object
38	sideEffect31	1329 non-null	object
39	sideEffect32	1328 non-null	object
40	sideEffect33	1169 non-null	object
41	sideEffect34	1166 non-null	object
42	sideEffect35	2 non-null	object
43	sideEffect36	2 non-null	object
44	sideEffect37	2 non-null	object
45	sideEffect38	2 non-null	object
46	sideEffect39	2 non-null	object
47	sideEffect40	2 non-null	object
48	sideEffect41	2 non-null	object
49	use0	248218 non-null	object
50	use1	73365 non-null	object
51	use2	28307 non-null	object
52	use3	7379 non-null	object
53	use4	4971 non-null	object
54	Chemical Class	137791 non-null	object

```
55 Habit Forming      248218 non-null object
56 Therapeutic Class  248149 non-null object
57 Action Class       138036 non-null object
dtypes: int64(1), object(57)
memory usage: 109.8+ MB
```

```
In [79]: df.columns
```

```
Out[79]: Index(['id', 'name', 'substitute0', 'substitute1', 'substitute2',
               'substitute3', 'substitute4', 'sideEffect0', 'sideEffect1',
               'sideEffect2', 'sideEffect3', 'sideEffect4', 'sideEffect5',
               'sideEffect6', 'sideEffect7', 'sideEffect8', 'sideEffect9',
               'sideEffect10', 'sideEffect11', 'sideEffect12', 'sideEffect13',
               'sideEffect14', 'sideEffect15', 'sideEffect16', 'sideEffect17',
               'sideEffect18', 'sideEffect19', 'sideEffect20', 'sideEffect21',
               'sideEffect22', 'sideEffect23', 'sideEffect24', 'sideEffect25',
               'sideEffect26', 'sideEffect27', 'sideEffect28', 'sideEffect29',
               'sideEffect30', 'sideEffect31', 'sideEffect32', 'sideEffect33',
               'sideEffect34', 'sideEffect35', 'sideEffect36', 'sideEffect37',
               'sideEffect38', 'sideEffect39', 'sideEffect40', 'sideEffect41', 'use0',
               'use1', 'use2', 'use3', 'use4', 'Chemical Class', 'Habit Forming',
               'Therapeutic Class', 'Action Class'],
              dtype='object')
```

4. Data Cleaning

```
In [80]: print(df.isnull().sum())
```

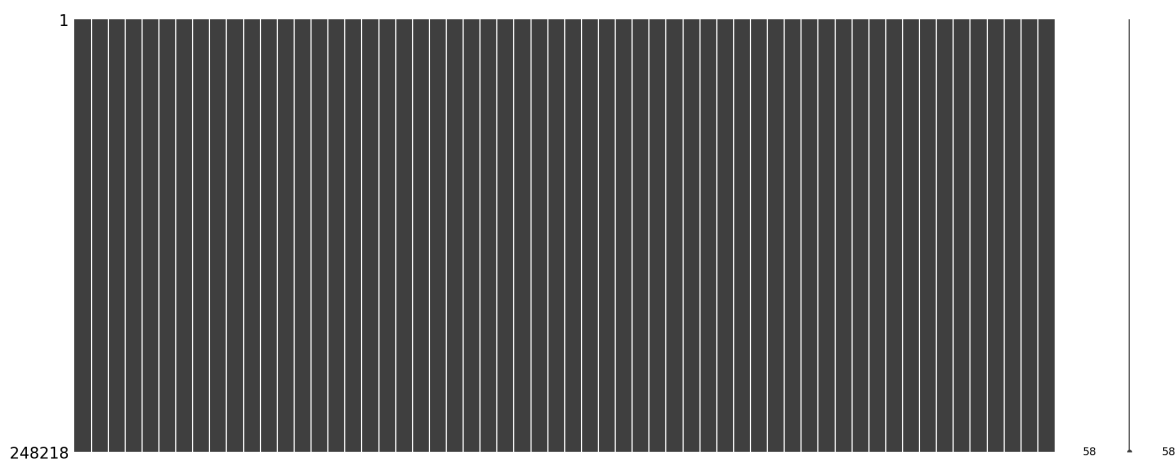
id	0
name	0
substitute0	9597
substitute1	14351
substitute2	17985
substitute3	21362
substitute4	24256
sideEffect0	0
sideEffect1	9802
sideEffect2	18718
sideEffect3	40580
sideEffect4	84658
sideEffect5	116960
sideEffect6	156361
sideEffect7	180468
sideEffect8	199712
sideEffect9	210510
sideEffect10	220944
sideEffect11	227887
sideEffect12	231936
sideEffect13	233491
sideEffect14	237799
sideEffect15	240537
sideEffect16	242209
sideEffect17	242836
sideEffect18	243703
sideEffect19	244272
sideEffect20	244995
sideEffect21	245093
sideEffect22	245170
sideEffect23	245313
sideEffect24	245495
sideEffect25	246715
sideEffect26	246715
sideEffect27	246724
sideEffect28	246724
sideEffect29	246780
sideEffect30	246889
sideEffect31	246889
sideEffect32	246890
sideEffect33	247049
sideEffect34	247052
sideEffect35	248216
sideEffect36	248216
sideEffect37	248216
sideEffect38	248216
sideEffect39	248216
sideEffect40	248216
sideEffect41	248216
use0	0
use1	174853
use2	219911
use3	240839
use4	243247
Chemical Class	110427
Habit Forming	0
Therapeutic Class	69
Action Class	110182
dtype:	int64

```
In [81]: msno.matrix(df)
plt.show()
```



```
In [82]: df.fillna('None', inplace=True)
```

```
In [83]: msno.matrix(df)
plt.show()
```



```
In [84]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 248218 entries, 0 to 248217
Data columns (total 58 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   id                    248218 non-null  int64
 1   name                  248218 non-null  object
 2   substitute0           248218 non-null  object
 3   substitute1           248218 non-null  object
 4   substitute2           248218 non-null  object
 5   substitute3           248218 non-null  object
 6   substitute4           248218 non-null  object
 7   sideEffect0           248218 non-null  object
 8   sideEffect1           248218 non-null  object
 9   sideEffect2           248218 non-null  object
10  sideEffect3           248218 non-null  object
11  sideEffect4           248218 non-null  object
12  sideEffect5           248218 non-null  object
13  sideEffect6           248218 non-null  object
14  sideEffect7           248218 non-null  object
15  sideEffect8           248218 non-null  object
16  sideEffect9           248218 non-null  object
17  sideEffect10          248218 non-null  object
18  sideEffect11          248218 non-null  object
19  sideEffect12          248218 non-null  object
20  sideEffect13          248218 non-null  object
21  sideEffect14          248218 non-null  object
22  sideEffect15          248218 non-null  object
23  sideEffect16          248218 non-null  object
24  sideEffect17          248218 non-null  object
25  sideEffect18          248218 non-null  object
26  sideEffect19          248218 non-null  object
27  sideEffect20          248218 non-null  object
28  sideEffect21          248218 non-null  object
29  sideEffect22          248218 non-null  object
30  sideEffect23          248218 non-null  object
31  sideEffect24          248218 non-null  object
32  sideEffect25          248218 non-null  object
33  sideEffect26          248218 non-null  object
34  sideEffect27          248218 non-null  object
35  sideEffect28          248218 non-null  object
36  sideEffect29          248218 non-null  object
37  sideEffect30          248218 non-null  object
38  sideEffect31          248218 non-null  object
39  sideEffect32          248218 non-null  object
40  sideEffect33          248218 non-null  object
41  sideEffect34          248218 non-null  object
42  sideEffect35          248218 non-null  object
43  sideEffect36          248218 non-null  object
44  sideEffect37          248218 non-null  object
45  sideEffect38          248218 non-null  object
46  sideEffect39          248218 non-null  object
47  sideEffect40          248218 non-null  object
48  sideEffect41          248218 non-null  object
49  use0                  248218 non-null  object
50  use1                  248218 non-null  object
51  use2                  248218 non-null  object
52  use3                  248218 non-null  object
53  use4                  248218 non-null  object
54  Chemical Class        248218 non-null  object

```

```
55 Habit Forming      248218 non-null object
56 Therapeutic Class  248218 non-null object
57 Action Class       248218 non-null object
dtypes: int64(1), object(57)
memory usage: 109.8+ MB
```

```
In [96]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 248218 entries, 0 to 248217
```

```
Data columns (total 64 columns):
```

#	Column	Non-Null Count	Dtype
0	id	248218 non-null	int64
1	name	248218 non-null	object
2	substitute0	248218 non-null	object
3	substitute1	248218 non-null	object
4	substitute2	248218 non-null	object
5	substitute3	248218 non-null	object
6	substitute4	248218 non-null	object
7	sideEffect0	248218 non-null	object
8	sideEffect1	248218 non-null	object
9	sideEffect2	248218 non-null	object
10	sideEffect3	248218 non-null	object
11	sideEffect4	248218 non-null	object
12	sideEffect5	248218 non-null	object
13	sideEffect6	248218 non-null	object
14	sideEffect7	248218 non-null	object
15	sideEffect8	248218 non-null	object
16	sideEffect9	248218 non-null	object
17	sideEffect10	248218 non-null	object
18	sideEffect11	248218 non-null	object
19	sideEffect12	248218 non-null	object
20	sideEffect13	248218 non-null	object
21	sideEffect14	248218 non-null	object
22	sideEffect15	248218 non-null	object
23	sideEffect16	248218 non-null	object
24	sideEffect17	248218 non-null	object
25	sideEffect18	248218 non-null	object
26	sideEffect19	248218 non-null	object
27	sideEffect20	248218 non-null	object
28	sideEffect21	248218 non-null	object
29	sideEffect22	248218 non-null	object
30	sideEffect23	248218 non-null	object
31	sideEffect24	248218 non-null	object
32	sideEffect25	248218 non-null	object
33	sideEffect26	248218 non-null	object
34	sideEffect27	248218 non-null	object
35	sideEffect28	248218 non-null	object
36	sideEffect29	248218 non-null	object
37	sideEffect30	248218 non-null	object
38	sideEffect31	248218 non-null	object
39	sideEffect32	248218 non-null	object
40	sideEffect33	248218 non-null	object
41	sideEffect34	248218 non-null	object
42	sideEffect35	248218 non-null	object
43	sideEffect36	248218 non-null	object
44	sideEffect37	248218 non-null	object
45	sideEffect38	248218 non-null	object
46	sideEffect39	248218 non-null	object
47	sideEffect40	248218 non-null	object
48	sideEffect41	248218 non-null	object
49	use0	248218 non-null	object
50	use1	248218 non-null	object
51	use2	248218 non-null	object
52	use3	248218 non-null	object
53	use4	248218 non-null	object
54	Chemical Class	248218 non-null	object

```

55 Habit Forming                248218 non-null object
56 Therapeutic Class            248218 non-null object
57 Action Class                 248218 non-null object
58 substitute_count             248218 non-null int64
59 side_effect_count            248218 non-null int64
60 Chemical Class Encoded       248218 non-null int32
61 Habit Forming Encoded        248218 non-null int32
62 Therapeutic Class Encoded    248218 non-null int32
63 Action Class Encoded         248218 non-null int32

```

dtypes: int32(4), int64(3), object(57)

memory usage: 117.4+ MB

5. Analyzing Substitutes

```

In [85]: substitute_columns = [f'substitute{i}' for i in range(5)]

df['substitute_count'] = df[substitute_columns].apply(lambda row: sum(cell != 'N'
print(df[['name', 'substitute_count']].head())

```

	name	substitute_count
0	augmentin 625 duo tablet	5
1	azithral 500 tablet	5
2	ascoril ls syrup	5
3	allegra 120mg tablet	5
4	avil 25 tablet	1

```

In [86]: substitute_distribution = df['substitute_count'].value_counts().sort_index()

print(substitute_distribution)

```

```

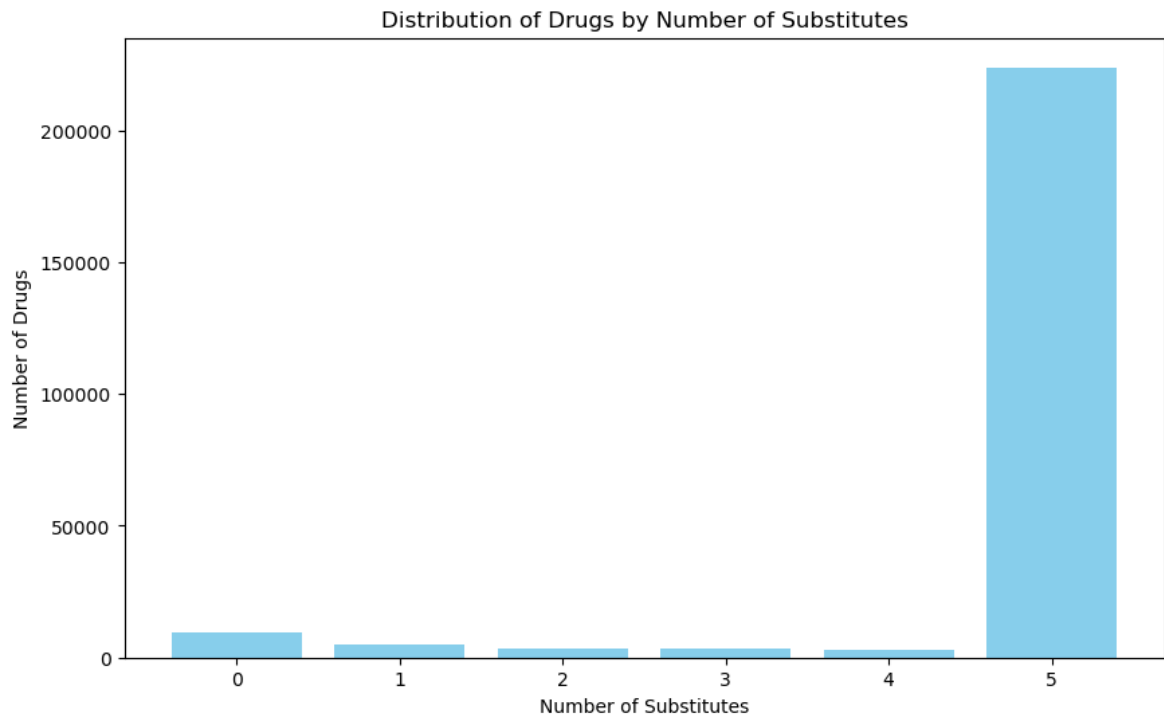
substitute_count
0      9597
1      4754
2      3634
3      3377
4       2894
5     223962
Name: count, dtype: int64

```

```

In [87]: plt.figure(figsize=(10, 6))
plt.bar(substitute_distribution.index, substitute_distribution.values, color='sk
plt.xlabel('Number of Substitutes')
plt.ylabel('Number of Drugs')
plt.title('Distribution of Drugs by Number of Substitutes')
plt.xticks(range(0, max(substitute_distribution.index) + 1))
plt.show()

```



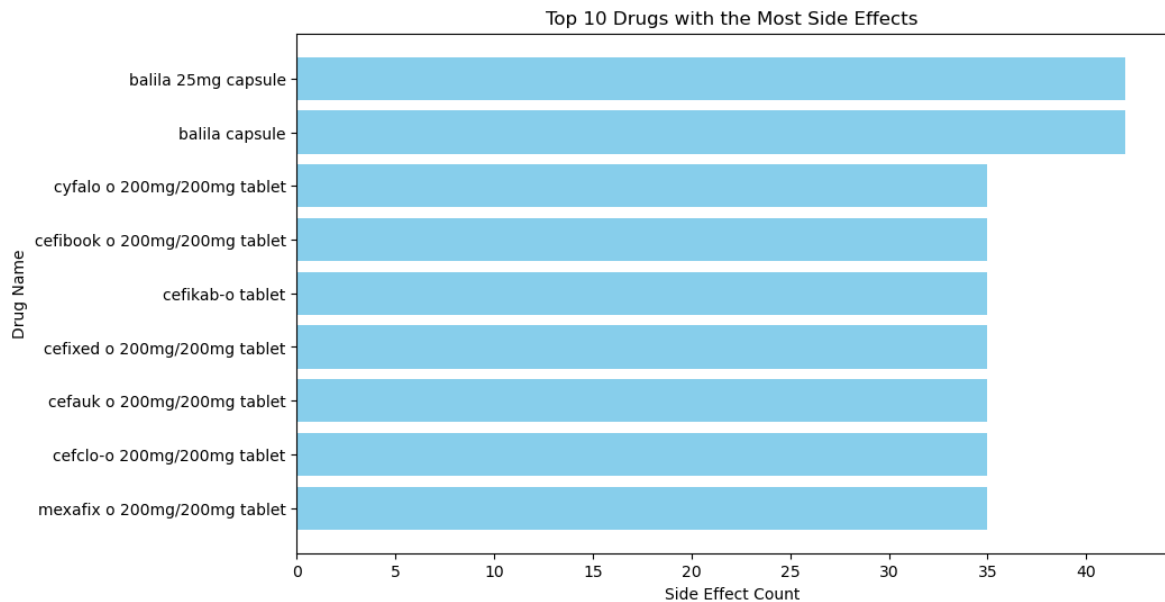
6. Analyzing Side Effects

```
In [88]: side_effect_columns = [f'sideEffect{i}' for i in range(42)]

df['side_effect_count'] = df[side_effect_columns].apply(lambda row: sum(cell !=
# Top 10 drugs with the most side effects
top_10_side_effects = df.sort_values('side_effect_count', ascending=False).head(
print(top_10_side_effects[['name', 'side_effect_count']])
```

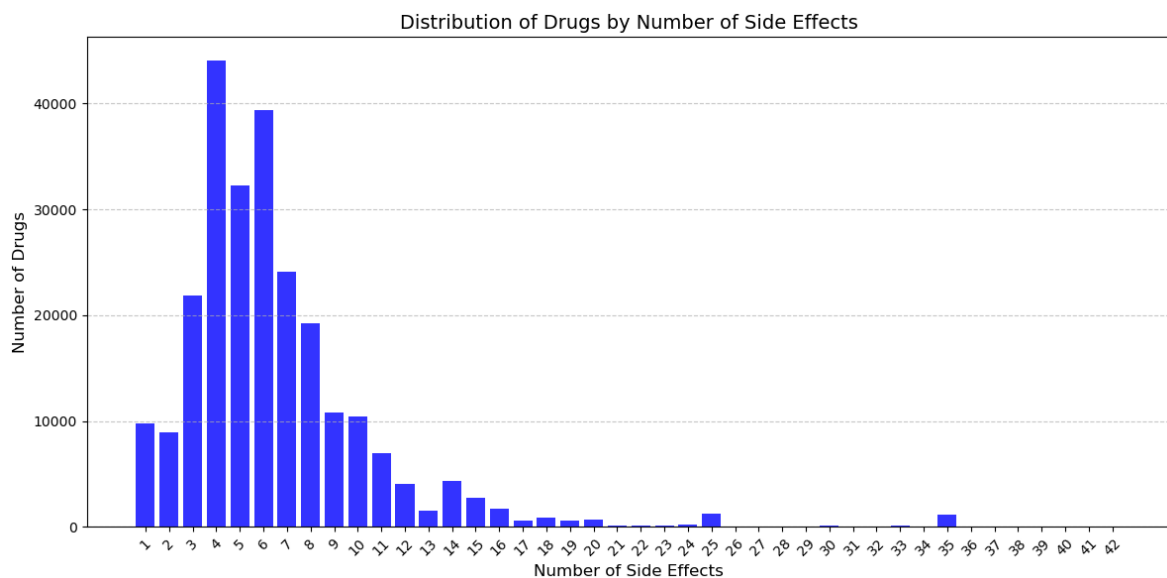
	name	side_effect_count
30832	balila 25mg capsule	42
30664	balila capsule	42
61995	cyfalo o 200mg/200mg tablet	35
62411	cefibook o 200mg/200mg tablet	35
62379	cefikab-o tablet	35
62346	cefixed o 200mg/200mg tablet	35
62208	cefauk o 200mg/200mg tablet	35
62194	cefauk o 200mg/200mg tablet	35
62190	cefclo-o 200mg/200mg tablet	35
159781	mexafix o 200mg/200mg tablet	35

```
In [63]: # Chart for drugs with the most side effects
plt.figure(figsize=(10, 6))
plt.barh(top_10_side_effects['name'], top_10_side_effects['side_effect_count'],
plt.xlabel('Side Effect Count')
plt.ylabel('Drug Name')
plt.title('Top 10 Drugs with the Most Side Effects')
plt.gca().invert_yaxis()
plt.show()
```



```
In [89]: side_effect_distribution = df['side_effect_count'].value_counts().sort_index()

plt.figure(figsize=(12, 6))
plt.bar(side_effect_distribution.index, side_effect_distribution.values, color='blue')
plt.xlabel('Number of Side Effects', fontsize=12)
plt.ylabel('Number of Drugs', fontsize=12)
plt.title('Distribution of Drugs by Number of Side Effects', fontsize=14)
plt.xticks(range(min(side_effect_distribution.index), max(side_effect_distribution.index)+1),
            labels=range(min(side_effect_distribution.index), max(side_effect_distribution.index)+1),
            alpha=0.7)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



```
In [90]: from collections import Counter

side_effect_columns = [f'sideEffect{i}' for i in range(42)]
all_side_effects = df[side_effect_columns].values.flatten()
filtered_side_effects = [effect for effect in all_side_effects if effect != 'None']

side_effect_counts = Counter(filtered_side_effects)
most_common_side_effects = side_effect_counts.most_common(10)
print("İlk 10 en yaygın yan etki:")
```

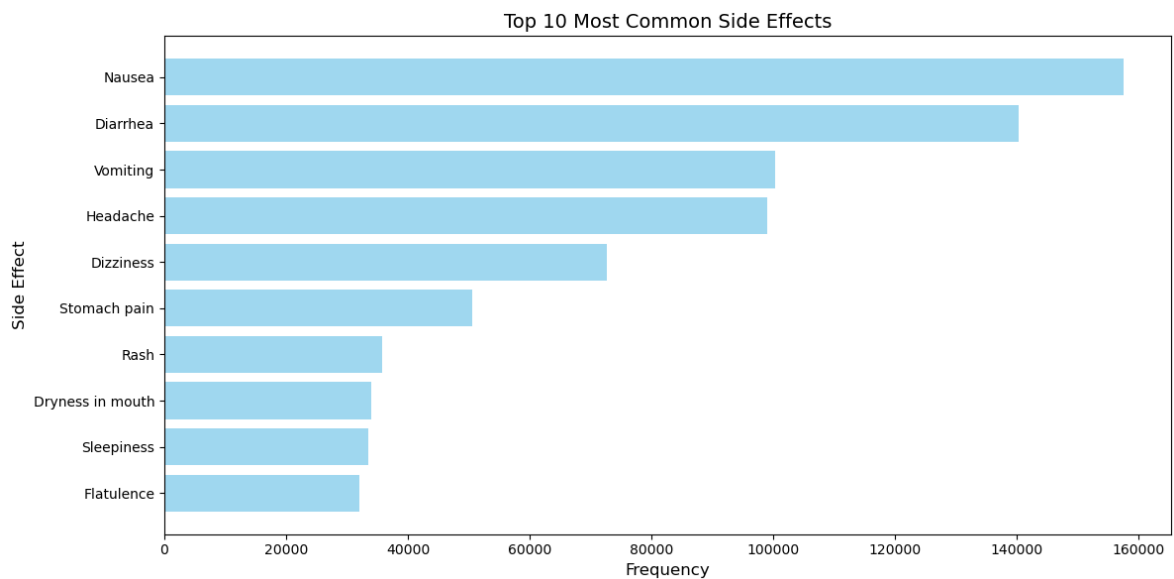
```
for effect, count in most_common_side_effects:
    print(f"{effect}: {count}")
```

İlk 10 en yaygın yan etki:

Nausea: 157428
 Diarrhea: 140295
 Vomiting: 100331
 Headache: 99015
 Dizziness: 72637
 Stomach pain: 50603
 Rash: 35749
 Dryness in mouth: 33984
 Sleepiness: 33516
 Flatulence: 32045

```
In [66]: effects, counts = zip(*most_common_side_effects)

plt.figure(figsize=(12, 6))
plt.barh(effects, counts, color='skyblue', alpha=0.8)
plt.xlabel('Frequency', fontsize=12)
plt.ylabel('Side Effect', fontsize=12)
plt.title('Top 10 Most Common Side Effects', fontsize=14)
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()
```



7. Classification Analysis

```
In [91]: # Group by Chemical Class and analyze side effects
chemical_class_effects = df.groupby('Chemical Class')[side_effects_columns].apply
print(chemical_class_effects.sort_values(ascending=False).head(10))
```

```

Chemical Class
None 4637934
Fluoroquinolone 333018
Broad Spectrum (Third & fourth generation cephalosporins) 310884
Macrolides 235368
Sulfinylbenzimidazole Derivative 204582
Broad spectrum (Third & fourth generation cephalosporins) 188664
Glucocorticoids, progestogens and derivatives 123564
Azoles {Triazoles} 114828
Intermediate spectrum {Second generation cephalosporins} 112140
Carbazole Derivative 107814
dtype: int64

```

```

In [92]: therapeutic_class_effects = df.groupby('Therapeutic Class')[side_effects_columns]
print(therapeutic_class_effects.sort_values(ascending=False).head(10))

```

```

Therapeutic Class
ANTI INFECTIVES 2415126
GASTRO INTESTINAL 1412460
PAIN ANALGESICS 1355214
NEURO CNS 979398
RESPIRATORY 968184
CARDIAC 772002
ANTI DIABETIC 490518
OPHTHAL 444066
DERMA 415086
HORMONES 236418
dtype: int64

```

```

In [93]: action_class_effects = df.groupby('Action Class')[side_effects_columns].apply(lambda x: x.sort_values(ascending=False).head(10))
print(action_class_effects.sort_values(ascending=False).head(10))

```

```

Action Class
None 4627644
Cephalosporins: 3 generation 526890
Quinolones/ Fluroquinolones 333186
Fungal ergosterol synthesis inhibitor 282702
Proton pump inhibitors 267918
Macrolides 249228
Glucocorticoids 223020
H1 Antihistaminics (second Generation) 169134
HMG CoA inhibitors (statins) 120288
Cephalosporins: 2nd generation 115542
dtype: int64

```

```

In [95]: from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()
df['Chemical Class Encoded'] = label_encoder.fit_transform(df['Chemical Class'])
df['Habit Forming Encoded'] = label_encoder.fit_transform(df['Habit Forming'])
df['Therapeutic Class Encoded'] = label_encoder.fit_transform(df['Therapeutic Class'])
df['Action Class Encoded'] = label_encoder.fit_transform(df['Action Class'])

```

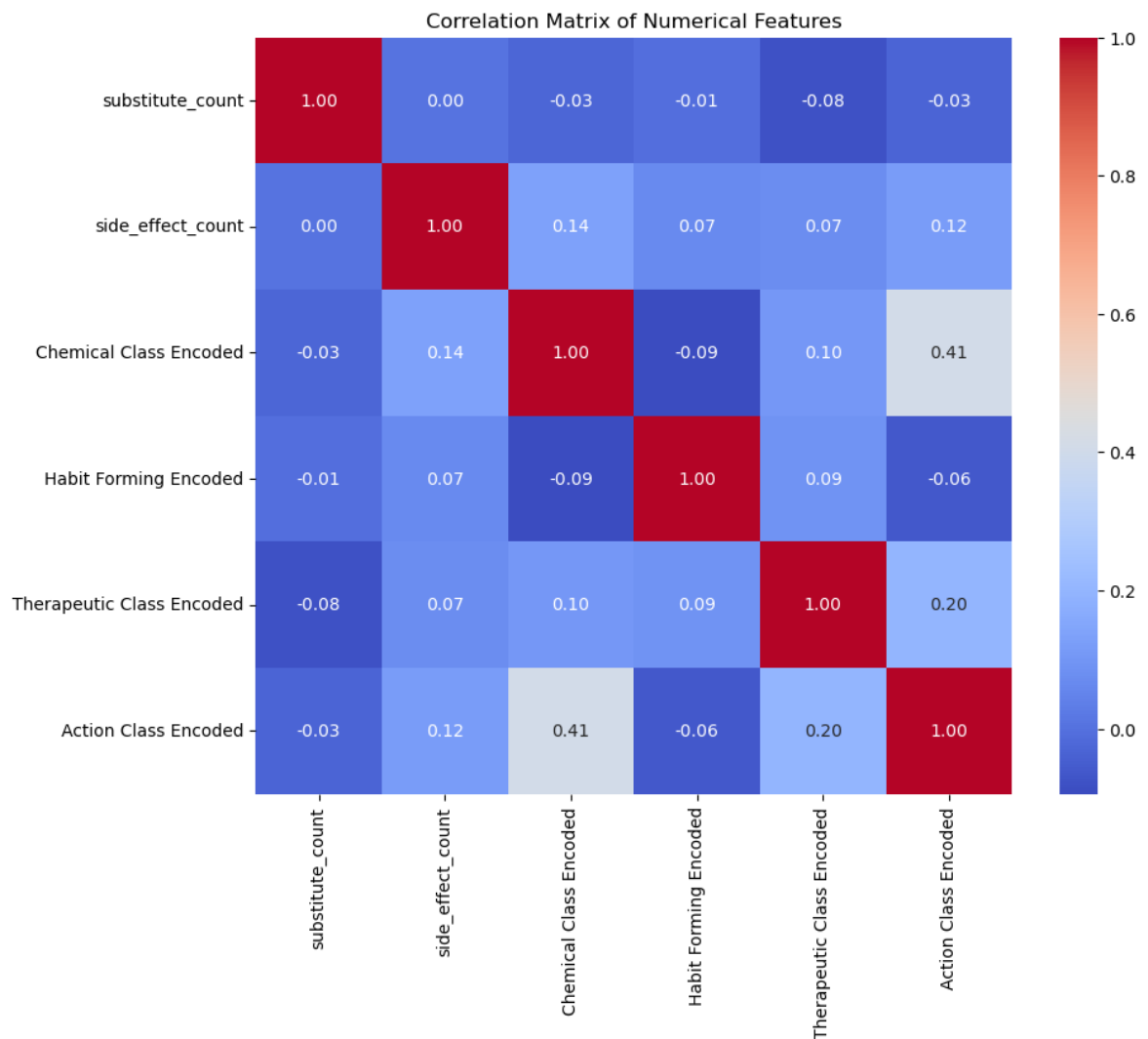
```

In [97]: numeric_columns = ['substitute_count', 'side_effect_count',
                             'Chemical Class Encoded', 'Habit Forming Encoded',
                             'Therapeutic Class Encoded', 'Action Class Encoded']

correlation_matrix = df[numeric_columns].corr()

```

```
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', square=True)
plt.title('Correlation Matrix of Numerical Features')
plt.show()
```



8. Clustering analysis

```
In [101... clustering_data = df[['Chemical Class Encoded', 'Therapeutic Class Encoded', 'side_effect_count']]
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
clustering_data_scaled = scaler.fit_transform(clustering_data)
```

```
In [102... from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=5, random_state=42)
clusters = kmeans.fit_predict(clustering_data_scaled)

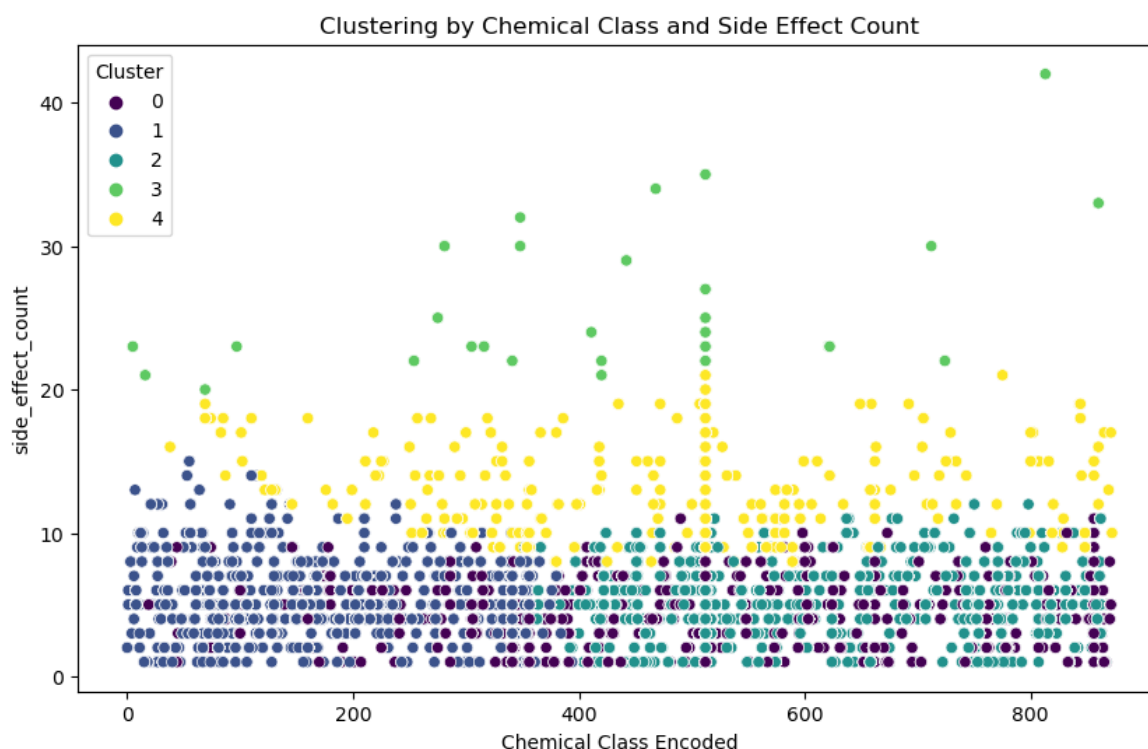
df['Cluster'] = clusters
```

C:\Users\user\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: Future Warning: The default value of `n\_init` will change from 10 to 'auto' in 1.4. Set the value of `n\_init` explicitly to suppress the warning

```
super()._check_params_vs_input(X, default_n_init=10)
```

```
In [103... plt.figure(figsize=(10, 6))
sns.scatterplot(x='Chemical Class Encoded', y='side_effect_count', hue='Cluster')
```

```
plt.title('Clustering by Chemical Class and Side Effect Count')  
plt.show()
```



Conclusion and Evaluation

In this analysis, the relationships between the areas of use, side effects and alternatives of drugs were examined in detail. Using a comprehensive dataset of more than 250,000 drugs, the following main findings were obtained:

Number of Alternatives:

It was observed that most drugs have at least one alternative. Alternatives are important in terms of drug variety and increase the accessibility options of patients.

Distribution of Side Effects:

Nausea, diarrhea and vomiting were among the most common side effects. This finding contributes to the understanding of the risks associated with widespread drug use.

Classification:

It was observed that there were significant differences between chemical and therapeutic classes. For example, anti-infective drugs have the widest coverage in terms of side effects.

Cluster Analysis:

Clusters formed according to chemical classes and number of side effects helped to understand the characteristics of drug groups. These results can be guiding in drug selection.

Future Studies

This analysis has the potential to make more informed choices in drug use and guide new drug development processes. In the future, such analyses can be expanded with patient data to develop personalized treatment recommendations. Additionally, studies can be conducted on predicting side effects with machine learning models.

In []: