**Project Title: "Basic Drug Effectiveness and Side Effects Analysis"**

**Project Overview:**

In this project, you'll analyze a **dataset of pharmaceutical drugs**, focusing on their **effectiveness** and **side effects**. You'll perform basic **data cleaning**, **exploratory data analysis (EDA)**, and **visualization** to identify patterns in drug side effects, effectiveness, and user ratings.

**Project Steps:**

**1. Dataset Selection:**

* **Option 1**: You can use a publicly available dataset such as the **Drugs.com dataset** or any dataset available on platforms like **Kaggle**. One example is a dataset with drug names, their side effects, effectiveness, and reviews.
* **Option 2**: You could create a simple dataset yourself with some sample drug data. Here's a small sample structure:

| **Drug Name** | **Effectiveness (1-5)** | **Side Effects** | **User Rating** |
| --- | --- | --- | --- |
| Paracetamol | 5 | Nausea, Dizziness | 4.5 |
| Ibuprofen | 4 | Headache, Stomach upset | 4.0 |
| Amoxicillin | 5 | Nausea, Vomiting | 4.2 |
| Aspirin | 3 | Stomach bleeding, Dizziness | 3.5 |
| Metformin | 4 | Nausea, Diarrhea | 3.8 |

**2. Data Cleaning:**

* Import the dataset into a **Pandas DataFrame**.
* Check for any missing or incorrect values, and handle them (e.g., filling missing values or dropping rows).
* Convert any textual data (like drug names) to lowercase or standardized formats for consistency.
* If your dataset includes user reviews, consider normalizing the text (e.g., removing stopwords, punctuation, and tokenizing).

**Example code snippet**:

python

Copy code

import pandas as pd

# Load the dataset

data = pd.read\_csv('drug\_data.csv')

# Check for missing values

print(data.isnull().sum())

# Fill missing values with a placeholder or drop them

data['Side Effects'].fillna('No side effects', inplace=True)

# Clean text (if applicable)

data['Drug Name'] = data['Drug Name'].str.lower()

**3. Exploratory Data Analysis (EDA):**

* Get basic insights into the dataset using **summary statistics**.
* Analyze the distribution of **effectiveness scores** and **user ratings**.
* Group by drug names and calculate average ratings or effectiveness scores to identify the most popular or effective drugs.

**Example code for basic EDA**:

python

Copy code

# Summary statistics

print(data.describe())

# Average effectiveness by drug

effectiveness\_by\_drug = data.groupby('Drug Name')['Effectiveness (1-5)'].mean()

print(effectiveness\_by\_drug)

# Check for the distribution of user ratings

print(data['User Rating'].value\_counts())

**4. Visualization:**

* Use **Matplotlib** or **Seaborn** to create simple visualizations such as:
  + **Bar plot** of average drug effectiveness.
  + **Pie chart** showing the most common side effects.
  + **Box plot** for user ratings distribution.

**Example code for visualizations**:

python

Copy code

import matplotlib.pyplot as plt

import seaborn as sns

# Bar plot for average drug effectiveness

effectiveness\_by\_drug.plot(kind='bar', color='skyblue', figsize=(10, 6))

plt.title('Average Drug Effectiveness')

plt.ylabel('Average Effectiveness')

plt.xlabel('Drug Name')

plt.xticks(rotation=45)

plt.show()

# Pie chart for side effects distribution

side\_effects\_count = data['Side Effects'].value\_counts()

side\_effects\_count.plot(kind='pie', autopct='%1.1f%%', figsize=(8, 8))

plt.title('Distribution of Side Effects')

plt.ylabel('')

plt.show()

# Box plot for user ratings distribution

sns.boxplot(x=data['User Rating'], color='lightgreen')

plt.title('User Rating Distribution')

plt.show()

**5. Conclusion:**

* Summarize your findings. For example, you may notice that certain drugs have higher effectiveness ratings, or you could identify which side effects are most common.
* You can also comment on which drugs might be better choices based on the **user ratings** and **effectiveness scores**.

**Final Deliverables:**

1. **Data Cleaning Script**: A Python script for cleaning and preprocessing the drug data.
2. **Exploratory Data Analysis**: Insights into the dataset, including averages, distributions, and groupings.
3. **Data Visualizations**: Bar charts, pie charts, and box plots to visualize drug effectiveness, side effects, and ratings.
4. **Report/Conclusion**: A summary of your findings from the analysis.

**How to Present This on Your Resume:**

You can describe the project like this:

**Project Name**: Basic Drug Effectiveness and Side Effects Analysis  
**Technologies Used**: Python, Pandas, Matplotlib, Seaborn  
**Description**:

* Cleaned and preprocessed a dataset of pharmaceutical drugs to handle missing values and standardized text.
* Performed **exploratory data analysis** (EDA) to derive insights on drug effectiveness, side effects, and user ratings.
* Created **visualizations** to depict the distribution of effectiveness scores and common side effects.
* Concluded key insights such as most effective drugs and common side effects based on data analysis.

**Link**: GitHub Repository

This project is **simpler** than the more complex recommendation system, but it still demonstrates important skills like **data analysis**, **cleaning**, and **visualization**—all of which are highly valuable for a **BPharmacy** graduate looking to work in **data science** or **healthcare analytics**.

This will be a great addition to your resume as it aligns both your **domain knowledge in pharmacy** and **data skills**.

Me : no thats not my opinion im getting side effect of each drug is one even it has two or more side effects

**ChatGPT said:**

ChatGPT

Ah, I see! You're mentioning that when plotting the pie chart, you're still getting the **count of each drug as 1**, even though some drugs have **multiple side effects**. This is because you're treating each row as a separate "unit" rather than counting the individual side effects properly.

Let's break this down and fix the issue.

**Issue:**

The problem is likely occurring because the **side effects column** contains multiple side effects (separated by commas), and you're counting each **drug** only once, instead of counting each **side effect** individually.

**Solution:**

You need to **split the side effects** column by commas and **count the frequency of each side effect** across all drugs. This way, each side effect is treated as an individual entity, even if it appears for multiple drugs.

**Updated Code:**

Here's a corrected version of the code that ensures you're counting **individual side effects** across all drugs, even if multiple side effects exist for one drug.

python

Copy code

import pandas as pd

import matplotlib.pyplot as plt

# Sample Data (the same as before)

data = {

'Drug Name': ['Paracetamol', 'Ibuprofen', 'Amoxicillin', 'Aspirin', 'Metformin', 'Fluoxetine',

'Hydrocodone', 'Omeprazole', 'Diphenhydramine', 'Loratadine', 'Ciprofloxacin',

'Penicillin', 'Lidocaine', 'Lorazepam'],

'Effectiveness (1-5)': [5, 4, 5, 3, 4, 4, 5, 5, 3, 4, 4, 5, 5, 3],

'Side Effects': ['Nausea, Dizziness', 'Headache, Stomach upset', 'Nausea, Vomiting', 'Stomach bleeding, Dizziness',

'Nausea, Diarrhea', 'Insomnia, Nausea', 'Drowsiness, Constipation', 'Headache, Diarrhea',

'Drowsiness, Dry mouth', 'Dry mouth, Fatigue', 'Nausea, Diarrhea, Dizziness', 'Rash, Diarrhea',

'Local irritation, Drowsiness', 'Drowsiness, Memory issues'],

'User Rating': [4.5, 4.0, 4.2, 3.5, 3.8, 4.3, 4.7, 4.6, 3.9, 4.2, 4.1, 4.4, 4.8, 3.6]

}

# Convert to DataFrame

df = pd.DataFrame(data)

# Split the side effects into individual items and count their occurrences

side\_effects\_split = df['Side Effects'].str.split(', ').explode() # Split and explode into rows

side\_effect\_counts = side\_effects\_split.value\_counts() # Count frequency of each side effect

# Plot the pie chart for the distribution of side effects

side\_effect\_counts.plot(kind='pie', autopct='%1.1f%%', figsize=(8, 8), colors=['#ff9999','#66b3ff','#99ff99','#ffcc99','#c2c2f0', '#ffb3e6'])

plt.title('Distribution of Side Effects')

plt.ylabel('') # Hide the y-label for a cleaner look

plt.show()

**Explanation of the Code:**

1. **str.split(', ')**: This splits the **side effects** column into individual side effects. For example, "Nausea, Dizziness" becomes ['Nausea', 'Dizziness'].
2. **explode()**: This function **expands** the list of side effects into multiple rows. For example, if a drug has two side effects, it will now be treated as two separate rows.
3. **value\_counts()**: This counts the occurrence of each side effect across all drugs.
4. **Pie Chart**: The pie chart now shows the **distribution of individual side effects** across all the drugs.

**Expected Outcome:**

* If a side effect like **"Nausea"** appears in multiple drugs, it will be counted for each occurrence.
* The pie chart will now show how frequently each side effect appears across all drugs, rather than just counting each drug once.

**Example of the Data Handling:**

For example, if you have:

| **Drug Name** | **Side Effects** |
| --- | --- |
| Paracetamol | Nausea, Dizziness |
| Ibuprofen | Headache, Stomach upset |

After using str.split(', ') and explode(), the data would look like:

| **Drug Name** | **Side Effect** |
| --- | --- |
| Paracetamol | Nausea |
| Paracetamol | Dizziness |
| Ibuprofen | Headache |
| Ibuprofen | Stomach upset |

Now the count for each side effect will be accurate, and the pie chart will show the correct distribution.