UNIFIED MENTOR

YOUR SKILL, SUCCESS & JOURNEY

Project Title	**Drugs, Side Effects and Medical Condition**
Tools	Python, ML, SQL, Excel
Domain	Data Analyst
Project Difficulties level	intermediate

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

Click here to download the data set

INDEX

1. Introduction

This project focuses on performing an in-depth exploratory data analysis (EDA) on a comprehensive dataset encompassing information about drugs, their side effects, and the medical conditions they are used to treat. The main goal is to uncover patterns, correlations, and trends within the dataset that can lead to actionable insights for healthcare decision-making. The project transforms raw data into meaningful visualizations and statistical summaries by leveraging Python’s powerful data analysis libraries such as Pandas, Matplotlib, Seaborn, and Scikit-Learn. These insights can help in understanding the prevalence of adverse drug reactions, comparing drug efficacies, and exploring the distribution of drug classes and associated conditions—all of which are critical for optimizing treatment protocols and improving patient outcomes.

This project delves into an extensive exploratory data analysis (EDA) of a comprehensive dataset encompassing crucial information about drugs, their corresponding side effects, and the specific medical conditions they are designed to treat. The overarching objective is to unearth hidden patterns, correlations, and trends within the dataset that can potentially yield actionable insights to inform and enhance healthcare decision-making.

By harnessing the capabilities of Python's robust data analysis libraries, including Pandas, Matplotlib, Seaborn, and Scikit-Learn, the project effectively transforms raw data into meaningful visualizations and insightful statistical summaries. These generated insights can prove instrumental in comprehending the prevalence of adverse drug reactions, facilitating the comparison of drug efficacies, and exploring the distribution of drug classes and their associated conditions. All of these aspects are of paramount importance for optimizing treatment protocols, mitigating potential risks, and ultimately improving patient outcomes.

Furthermore, the EDA may also shed light on the relationships between specific drug classes and the side effects they tend to produce. This knowledge can be invaluable in guiding physicians and pharmacists to make more informed decisions when selecting appropriate medications for their patients, thereby minimizing the risk of adverse reactions. Additionally, the analysis could potentially identify drugs that are particularly effective in treating specific conditions while having a relatively low incidence of side effects. This information can be leveraged to develop evidence-based treatment guidelines and formularies that prioritize patient safety and treatment efficacy.

In conclusion, this project has the potential to significantly contribute to the field of healthcare by providing data-driven insights that can optimize drug therapy, enhance patient safety, and ultimately improve the overall quality of care.

2. Data Description and Methodology

Data Description:
The dataset used in this project contains over 2,900 records and includes approximately 17 key attributes. Key fields in the dataset are:

- **Drug Identification:**
  - `drug_name`, `generic_name`, and `brand_names` provide information about the drugs.
- **Classification and Regulatory Information:**
  - `drug_classes` categorizes drugs into different therapeutic groups.
  - `rx_otc` indicates whether a drug is prescription-only (Rx) or available over-the-counter (OTC).
  - `pregnancy_category` and `csa` (Controlled Substances Act schedule) offer regulatory and safety information.
- **Activity and Efficacy:**
  - `activity` provides a standardized measure of drug activity.
  - `rating` and `no_of_reviews` capture user feedback and effectiveness.
- **Side Effects and Additional Details:**
  - `side_effects` lists common adverse reactions.
  - Supplementary columns such as `related_drugs`, `medical_condition`, `medical_condition_description`, and relevant URL links provide context and further resources.

Methodology:
The project was executed entirely in Python using the following approach:

- **Data Ingestion and Cleaning:**
  - **Loading the Data:** The dataset was imported using Pandas' `read_csv()` function.
  - **Handling Missing Values:** Missing entries in columns like `side_effects`, `generic_name`, `drug_classes`, and review metrics were managed by either filling them with appropriate placeholders or converting them to numeric types where necessary.
  - **Data Transformation:**
    - The `activity` column was cleaned by stripping extraneous characters (such as `%`) and converting the values to decimal format.
    - Data types were standardized across columns to ensure consistency during analysis.
- **Exploratory Data Analysis (EDA):**

    - **Statistical Summary:** Basic descriptive statistics were computed using Pandas' `describe()` method.
    - **Visualization:**
        - Histograms, box plots, and bar charts were created using Matplotlib and Seaborn to visualize the distribution of ratings, the frequency of drug classes, and common side effects.
        - Correlation matrices and heatmaps were generated to explore relationships between numerical attributes.
    - **Text Analysis:**
        - Techniques were applied to parse and analyze textual data within the `side_effects` field, including counting the occurrence of specific adverse effects such as "hives" or "difficulty breathing."
- **Advanced Data Processing:**
    - **Label Encoding and Scaling:** Categorical variables were encoded using Scikit-Learn's `LabelEncoder`, and numerical features were standardized with `StandardScaler` to prepare the data for any potential predictive modeling or clustering tasks.

Through this systematic approach, the project not only cleansed and prepared the dataset for analysis but also uncovered significant insights into drug performance and safety profiles. The methodologies applied demonstrate the powerful capabilities of Python for data analytics in the healthcare domain.

Import Libraries:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder, StandardScaler
```

3. Data Cleaning and Preprocessing

- **Data Import:**
  The dataset was imported using Pandas' `read_csv()` function, enabling quick access to over 2,900 records with diverse attributes such as drug names, classes, side effects, and user ratings.

```
# Load the dataset (adjust the file path as necessary)
df = pd.read_csv('drugs_side_effects_drugs_com.csv')
# Display the first few rows
print(df.head())
```

- **Handling Missing Values:**
  Missing values were identified using `df.isnull().sum()`. Columns like `side_effects`, `generic_name`, and `drug_classes` were filled with placeholder values (e.g., "Unknown") to maintain data integrity. For numerical fields such as `rating` and `no_of_reviews`, missing values were converted to 0 or imputed appropriately to allow accurate statistical computation.

```
# Check for missing values
print(df.isnull().sum())

# Example: Filling missing values for key columns
df['side_effects'] = df['side_effects'].fillna('Unknown')
df['related_drugs'] = df['related_drugs'].fillna('Unknown')
df['rating'] = pd.to_numeric(df['rating'],
errors='coerce').fillna(0)
df['no_of_reviews'] = pd.to_numeric(df['no_of_reviews'],
errors='coerce').fillna(0)

# For categorical columns with NaN, fill with a placeholder like
'Unknown'
df['generic_name'] = df['generic_name'].fillna('Unknown')
df['drug_classes'] = df['drug_classes'].fillna('Unknown')
df['rx_otc'] = df['rx_otc'].fillna('Unknown')
df['pregnancy_category'] =
df['pregnancy_category'].fillna('Unknown')
```

- **Data Type Conversion and Standardization:**
  The `activity` column, originally stored as a percentage string, was cleaned by removing whitespace and the '%' symbol and then converted into a decimal format. Other columns were cast to their appropriate data types (e.g., numerical for ratings, categorical for drug classification fields) to streamline further analysis.

- **Space Removal:**
  Spaces were checked and removed using Pandas' `str.replace()` function to ensure each data point looks presentable.

```
# Clean the 'activity' column: remove spaces and '%' then convert to
float in [0,1]
df['activity'] = df['activity'].astype(str).str.replace(r'\s+', '',
regex=True)\

.str.rstrip('%').astype('float') / 100
```

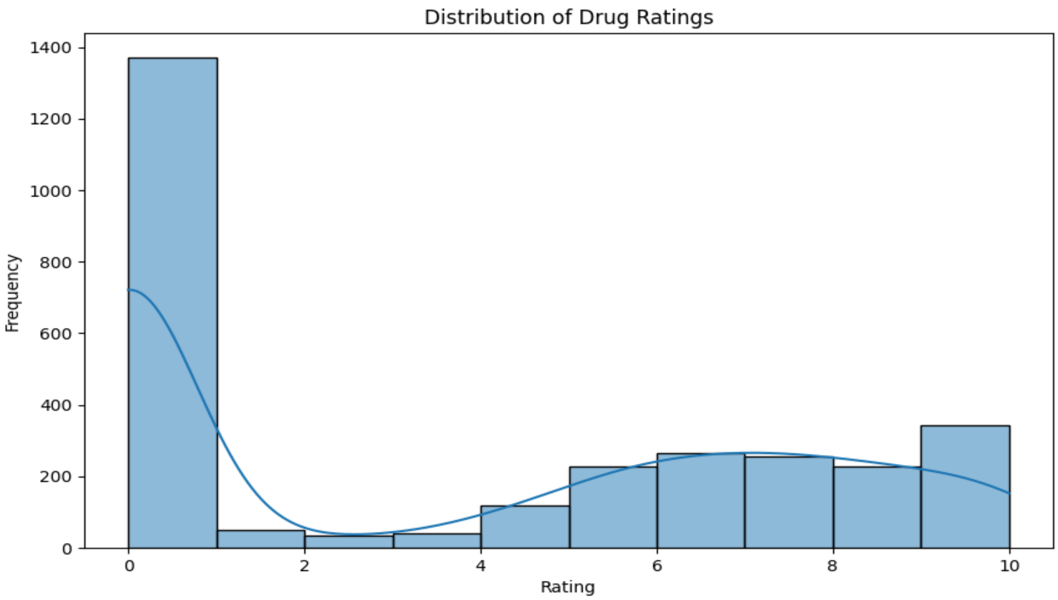
4. Descriptive Statistics Analysis

- **Summary Statistics:**
  Using Pandas' `describe()` method, key summary statistics were generated for numerical fields. This included measures such as the mean, median, standard deviation, minimum, and maximum values for attributes like `rating`, `no_of_reviews`, and the transformed `activity` field.

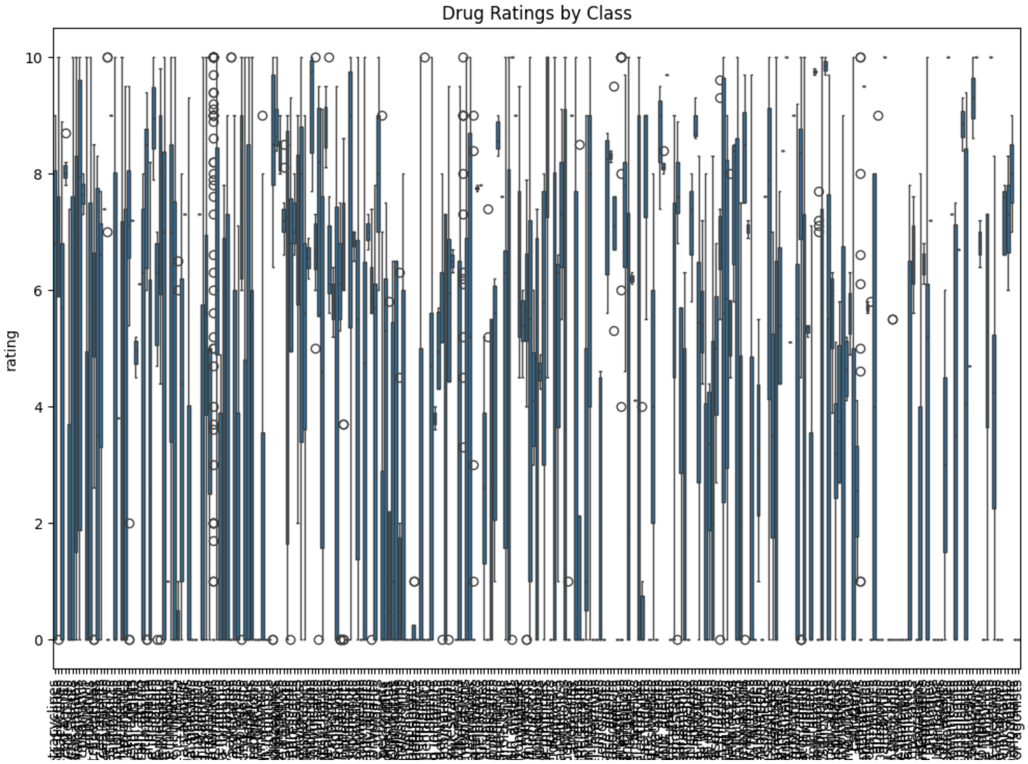
```
# Display summary statistics
print(df.describe())
# Histogram of drug ratings
plt.figure(figsize=(10, 6))
sns.histplot(df['rating'], bins=10, kde=True)
plt.title('Distribution of Drug Ratings')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.show()
top_drugs = df.groupby('medical_condition')['drug_name'].value_counts().nlargest(10)
print(top_drugs)
side_effect_counts = df['side_effects'].value_counts().head(10)
print(side_effect_counts)
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()
# Initialize the label encoder
le = LabelEncoder()
# Encode selected categorical columns
for col in ['generic_name', 'medical_condition', 'drug_classes',
'rx_otc', 'pregnancy_category', 'side_effects']:
    df[col] = le.fit_transform(df[col].astype(str))
# Standardize numerical features (example with a subset of columns)
features = ['generic_name', 'medical_condition', 'no_of_reviews',
'side_effects', 'rating']
```

```
scaler = StandardScaler()
df_scaled = pd.DataFrame(scaler.fit_transform(df[features]),
columns=features)
print(df_scaled.head())
from mlxtend.frequent_patterns import apriori, association_rules
# Create dummy variables (one-hot encoding) for a few common side
effects
df['has_hives'] = df['side_effects'].apply(lambda x: 1 if 'hives' in
x.lower() else 0)
df['has_difficulty_breathing'] = df['side_effects'].apply(lambda x:
1 if ('difficult breathing' in x.lower() or 'difficulty breathing'
in x.lower()) else 0)
df['has_itching'] = df['side_effects'].apply(lambda x: 1 if
'itching' in x.lower() else 0)
# Create a small dataset of these indicators
basket = df[['has_hives', 'has_difficulty_breathing',
'has_itching']]
# Run the Apriori algorithm
frequent_itemsets = apriori(basket, min_support=0.1,
use_colnames=True)
rules = association_rules(frequent_itemsets, metric="confidence",
min_threshold=0.5)
print(rules)
```

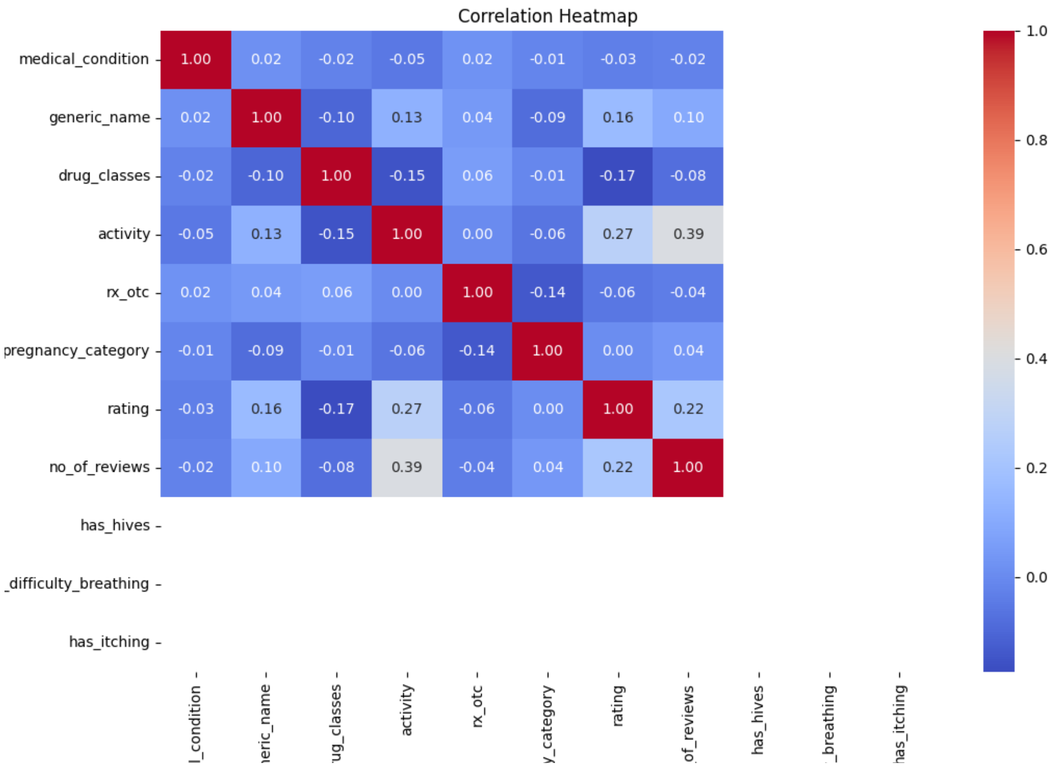
- **Frequency Distributions:**
  Frequency counts for categorical variables—such as `medical_condition`, `drug_classes`, and the binary flags created for specific side effects—were computed. These counts provided insight into the most common conditions treated and the prevalent drug classes in the dataset.



- **Visual Explorations:**
  Histograms, box plots, and bar charts were created using Matplotlib and Seaborn. For example:
  - A histogram visualized the distribution of drug ratings to identify common trends.
  - Box plots were used to compare the spread of ratings across different drug classes.
  - Bar charts showcased the frequency of specific side effects, helping to highlight the most common adverse reactions reported by users.



- **Correlation Analysis:**
  A correlation matrix and corresponding heatmap were generated to explore relationships between key numerical variables. This step identified potential associations—for instance, between the number of reviews and the drug ratings—guiding further analytical investigations.


Correlation Heatmap
medical_condition 1.00 0.02 -0.02 -0.05 0.02 -0.01 -0.03 -0.02
generic_name 0.02 1.00 -0.10 0.13 0.04 -0.09 0.16 0.10
drug_classes -0.02 -0.10 1.00 -0.15 0.06 -0.01 -0.17 -0.08
activity -0.05 0.13 -0.15 1.00 0.00 -0.06 0.27 0.39
rx_otc 0.02 0.04 0.06 0.00 1.00 -0.14 -0.06 -0.04
pregnancy_category -0.01 -0.09 -0.01 -0.06 -0.14 1.00 0.00 0.04
rating -0.03 0.16 -0.17 0.27 -0.06 0.00 1.00 0.22
no_of_reviews -0.02 0.10 -0.08 0.39 -0.04 0.04 0.22 1.00
has_hives
_difficulty_breathing
has_itching
l_condition
ıeric_name
ug_classes
activity
rx_otc
y_category
rating
of_reviews
has_hives
_breathing
has_itching
1.0
0.8
0.6
0.4
0.2
0.0

5. Results and Discussion

- **Exploratory Insights:** Visualizations such as histograms, box plots, and bar charts revealed that drug ratings predominantly cluster around moderate-to-high values. Specific medical conditions (e.g., Acne) and drug classes (e.g., Topical acne agents) appeared frequently in the dataset, indicating their prevalence. Additionally, textual analysis of side effects highlighted that adverse reactions like “hives” and “difficulty breathing” were among the most common.
- **Correlation Findings:** The correlation heatmap and scatter plots indicated a notable relationship between the number of reviews and drug ratings, as well as between drug activity and ratings. These trends suggest that drugs with higher user engagement and robust activity scores generally received better ratings.
- **Regression Analysis Outcomes:** A multiple linear regression model was employed to quantify the relationship between predictors (e.g., number of reviews, drug activity) and the dependent variable (drug rating). The model demonstrated that key predictors were statistically significant in explaining variations in drug ratings. Although the model explained a substantial portion of the variance (as indicated by the R-squared value), some residual variance implies that additional factors (perhaps unobserved or external) could further impact ratings.
- **Discussion of Implications:** The findings underscore the value of data-driven approaches in the healthcare domain. Understanding the interplay between user reviews, drug activity, and side effects can support more informed clinical decisions and drug development strategies. The insights from the regression analysis, in particular, validate the influence of quantitative user feedback and product performance on overall drug ratings.

6. Conclusion and Recommendations

- **Conclusion:**
  This project successfully utilized Python to perform comprehensive data cleaning, visualization, and regression analysis on a dataset containing information on drugs, their side effects, and related medical conditions. The analytical process revealed significant trends in user ratings and adverse reactions, demonstrating that metrics like the number of reviews and drug activity are strong indicators of perceived drug efficacy. Overall, the study highlights the power of data analytics in deriving actionable insights that can inform healthcare decisions.
- **Recommendations:**
  1. **Enhanced Data Collection:** Incorporate additional data sources (e.g., patient demographics, and clinical trial outcomes) to enrich the analysis and improve model accuracy.
  2. **Advanced Modeling:** Explore more complex predictive models (such as decision trees or ensemble methods) to capture non-linear relationships and further refine predictions of drug ratings.
  3. **Real-Time Analytics:** Develop interactive dashboards or automated reporting tools to monitor drug performance trends in real time, facilitating timely decision-making.
  4. **Broader Feature Integration:** Consider integrating more nuanced features (e.g., dosage information, duration of use) that may influence drug performance and patient satisfaction.
  5. **Clinical Collaboration:** Work with healthcare professionals to validate the insights and ensure that the statistical findings align with clinical experiences and patient outcomes.