**Classification of patient’s condition using Drug Review**



**Abstract**

Once a place for posting pretty websites to promote your business, the internet has now evolved to be a forum where consumers evaluate different products and services based on impressions and feedback from other like-minded consumers. In recent times, online reviews have created a new field in marketing and communication that bridges the gap between traditional word-of-mouth and a viral form of feedback that can influence consumer’s opinions. However, in the field of medicine, reviews made for drugs play an even more vital role as they can help in monitoring its adverse reactions and identify an overall impression of the drug among its users.

**Business Problem**

Patients and healthcare providers often rely on empirical evidence to gauge the effectiveness and side effects of medications. However, subjective patient experiences are equally important to understand the real-world impact of drugs. Pharmaceutical companies and healthcare providers face the challenge of synthesizing large volumes of unstructured patient feedback to inform drug development, improve patient care, and tailor treatments to individual needs.

**Objective**

The objective of analysing this dataset is to:

* Quantitatively assess overall patient satisfaction with various medications.
* Identify common themes in patient reviews that correlate with high or low drug ratings.
* Understand the impact of specific drugs on different medical conditions based on patient feedback.
* Provide actionable insights to healthcare providers and pharmaceutical companies to enhance patient care and drug development.

**Challenges**

* **Text Data Complexity:** The dataset consists of unstructured text data that requires significant preprocessing to extract meaningful insights. This includes cleaning HTML tags, handling special characters, and normalizing text.
* **High Dimensionality**: With text data, the feature space can become extremely large due to the vast number of unique words. Dimensionality reduction techniques may be necessary to manage this.
* **Subjectivity in Reviews**: Reviews are subjective and can vary greatly from person to person. This subjectivity can make it difficult to accurately classify sentiment or satisfaction.

**Real World Impact**

* **Drug Development**: Pharmaceutical companies can use the analysis to improve existing drugs or develop new ones that address common complaints or issues raised in patient reviews
* **Healthcare Policy**: Regulators and policymakers can use the data to monitor drug performance in the market, potentially leading to changes in drug approval processes or post-market surveillance.
* **Consumer Awareness**: Patients can benefit from the analysis by gaining access to aggregated patient experiences, helping them to make more informed decisions about their treatment options.
* **Market Trends**: Analysis of the dataset can reveal trends in medication usage and patient satisfaction, which can be valuable information for market analysts and investors in the healthcare sector.

**Dataset**

The drug review data set was collected from the **UCI machine learning** repository. The dataset provides patient reviews on specific drugs along with related conditions and a **10-star patient rating** reflecting overall patient satisfaction. The data is split into a **train (75%)** a **test (25%)** partition (see publication) and stored in two .csv (comm separated values) files, respectively.

**Data Fields**

The dataset consists of **161,297** entries and **7** columns. Here is the structure of the dataset:

1. **uniqueID(numerical):** An identifier for the review

2. **drugName (categorical):** name of drug

3. **condition (categorical):** name of condition

4. **review (text):** patient review

5. **rating (numerical):** 10-star patient rating

6. **date (date):** date of review entry

7. **usefulCount (numerical):** number of users who found review useful

**Performance Metrics**

Our task is a classification problem so we can use performance metrics like precision, recall, Accuracy and F1-score.

There are many metrics available to choose from:

1. Accuracy
2. Precision
3. Recall
4. F1 score

**Metric selection and reasoning:**

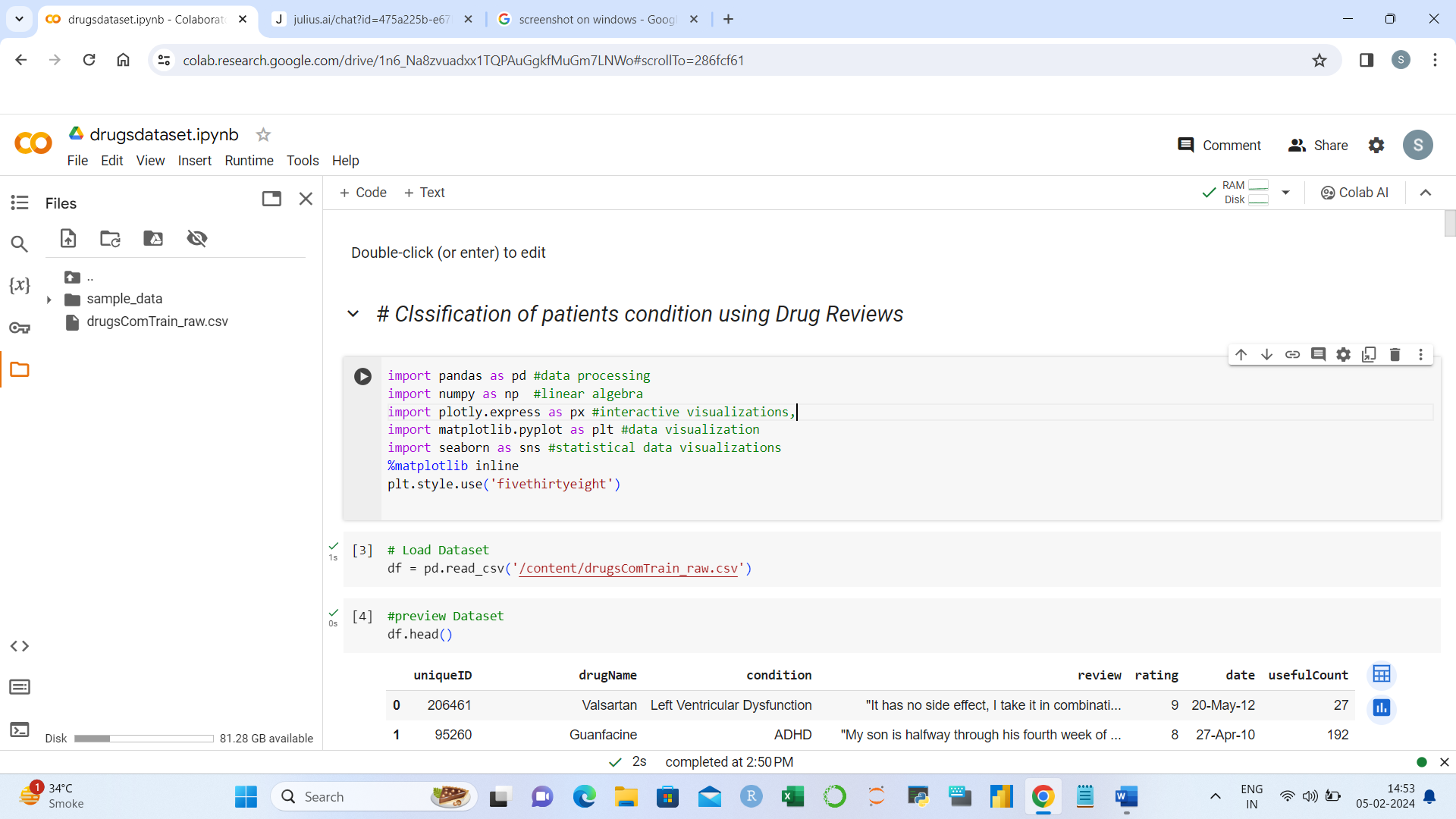
1. **Accuracy**: Accuracy measures the ratio of correctly predicted instances to the total instances in the dataset. It is a fundamental metric for assessing model performance.
2. **Sensitivity (Recall)**: Measures the proportion of correctly predicted positive observations to the all observations in actual class.
3. **Precision**: Precision assesses the ratio of correctly predicted positive observations to the total predicted positive observations.
4. **F1-Score**: It's the harmonic mean of precision and recall, offering a balance between the two metrics.

**Approach and Problem Type**

Since this is a classification problem and the data is sentiment analysis we can use machine learning and deep learning as well. The models are Multinomial Naïve Bayes, and Passive Aggressive Classifiers to name a few.

**Exploratory Data Analysis (EDA):**

Importing all the required libraries needed for the data



**Python**: A programming language well-suited for data analysis.

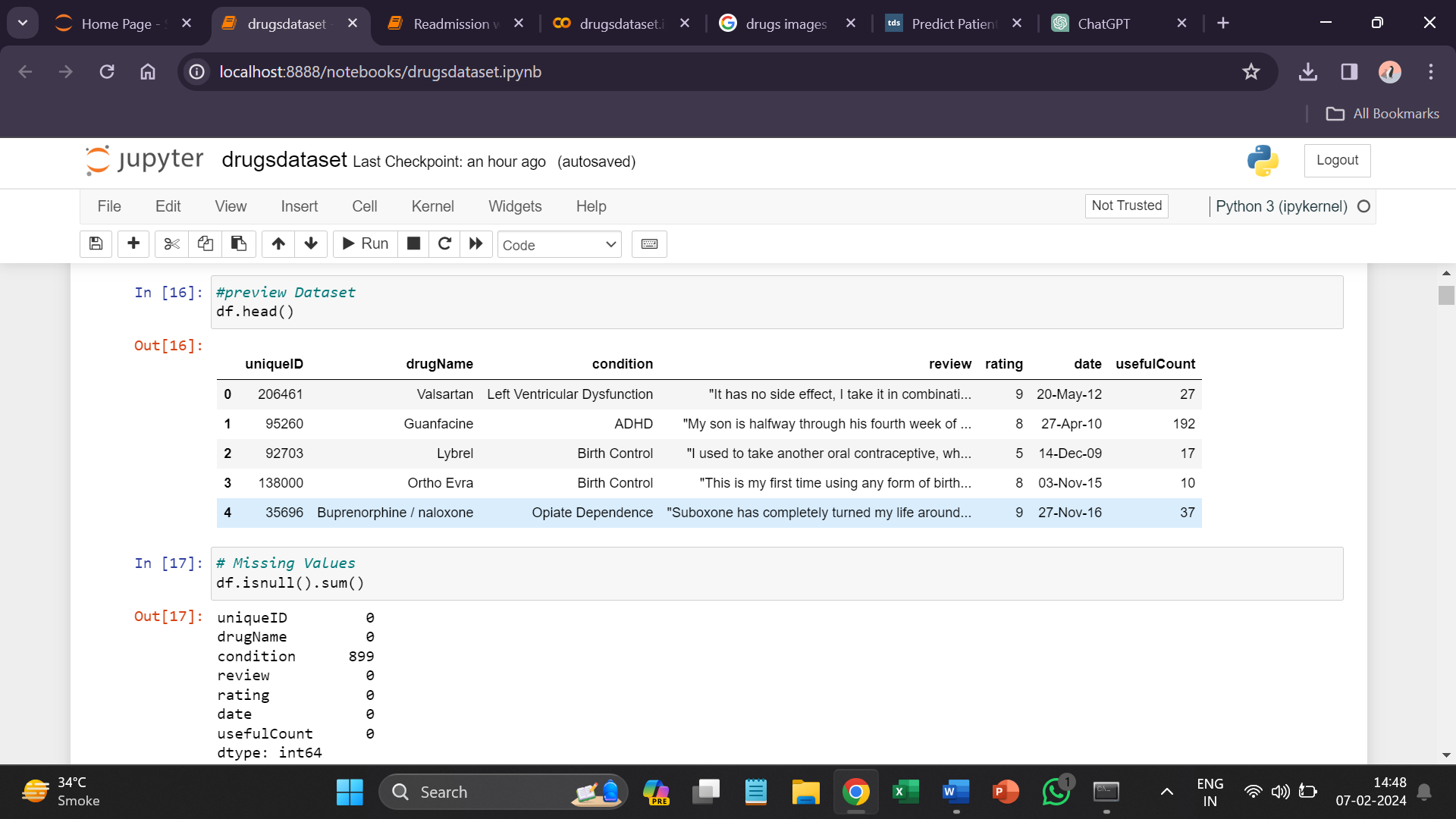
**Pandas**: A Python library for data manipulation and analysis.

**NumPy**: A library for numerical computing in Python.

**Scikit-learn**: A machine learning library in Python.

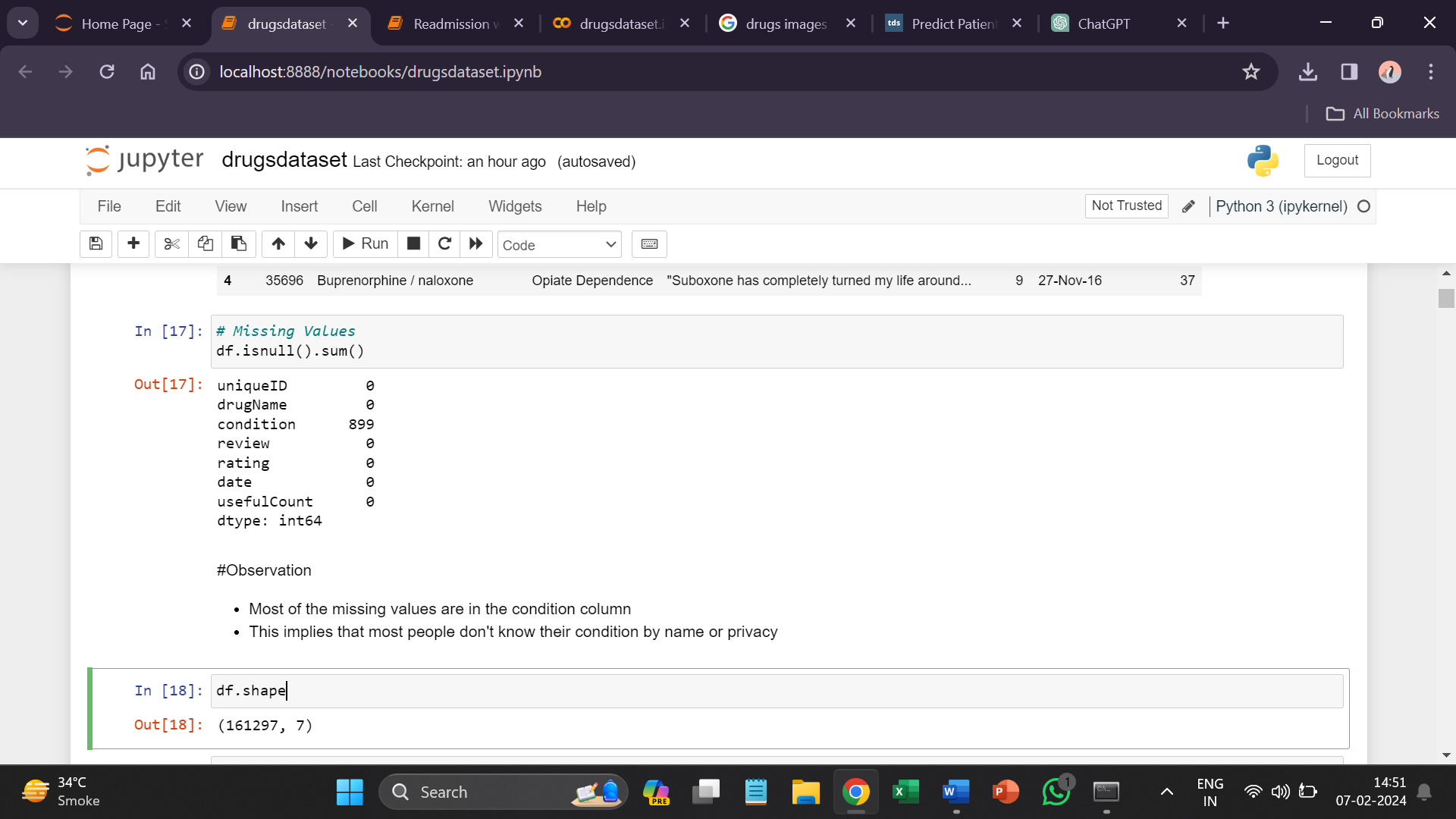
**Matplotlib/Seaborn**: Libraries for creating static, interactive, and informative visualizations in Python.

**## The first 5 rows:**

****

There are **161297** rows and **7** columns in the data.

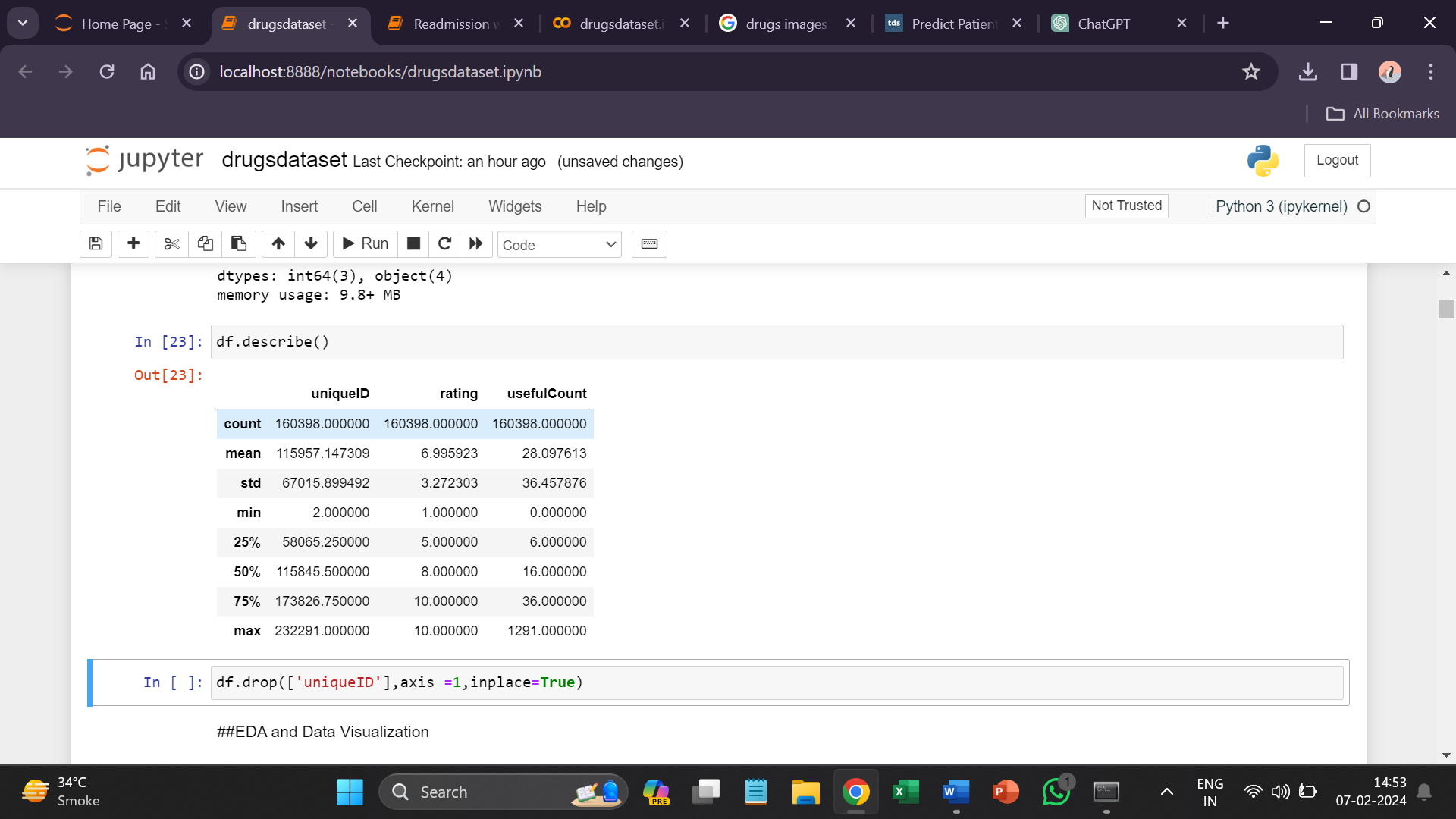
**## Checking the null values:**

****

Null values are only present in the ‘**Condition’** column i.e. 899

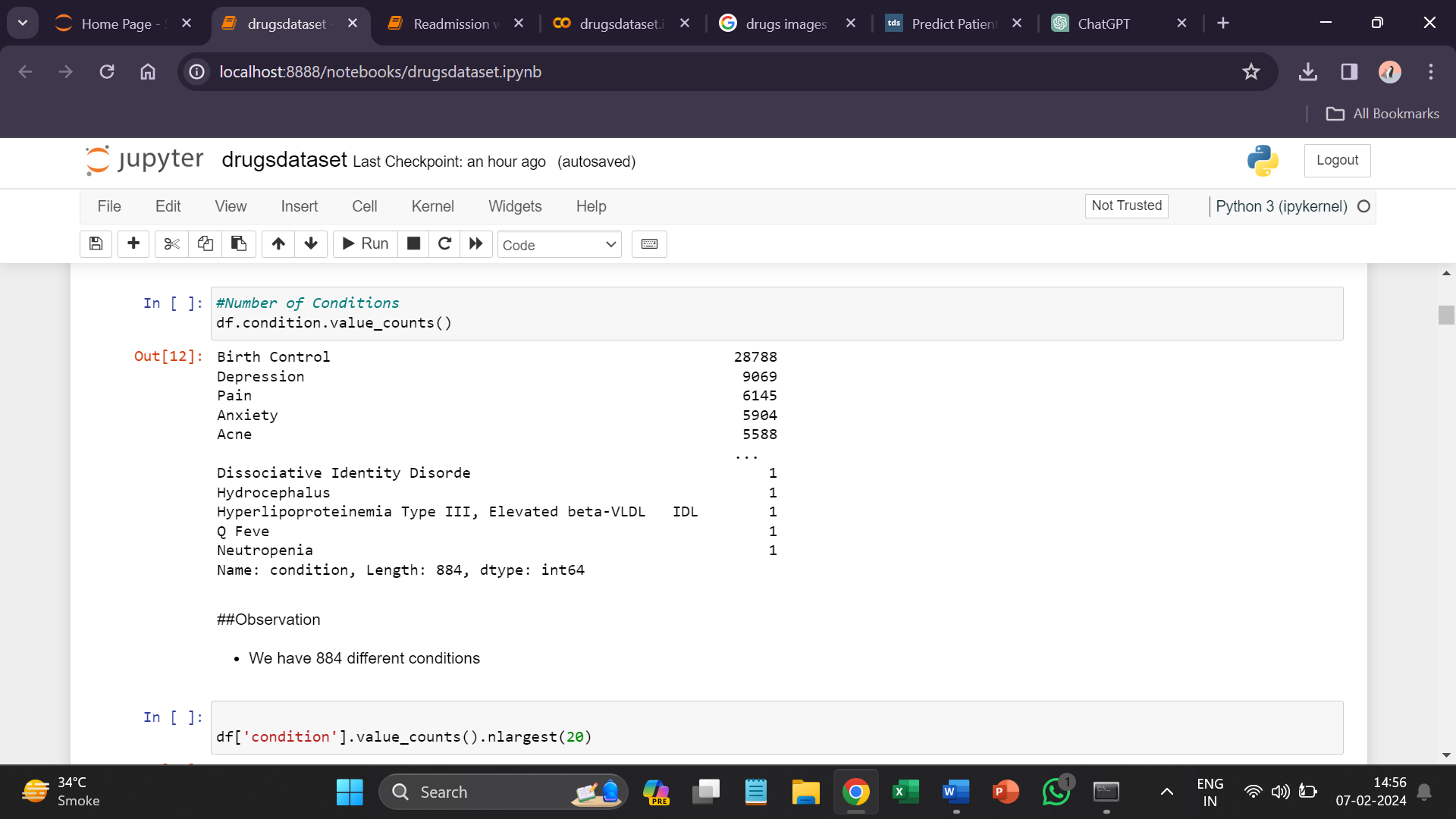
<

**## Looking at statistical description**

****

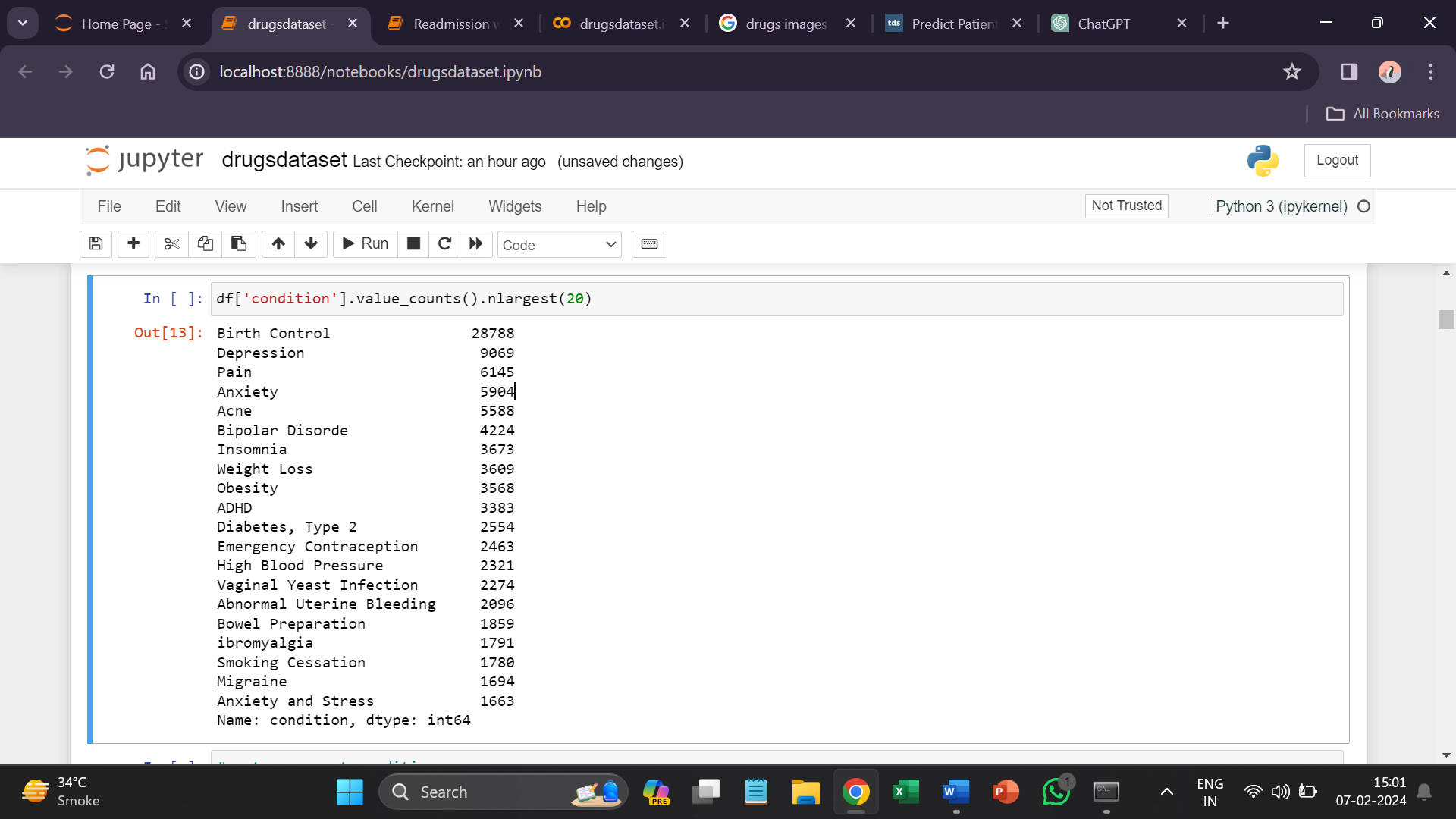
From above, we can see that maximum rating a drug has been given is **10** and the minimum is **1**. The number of users who had found review useful is **28** which is average.

**## Counting the number of unique conditions in data**

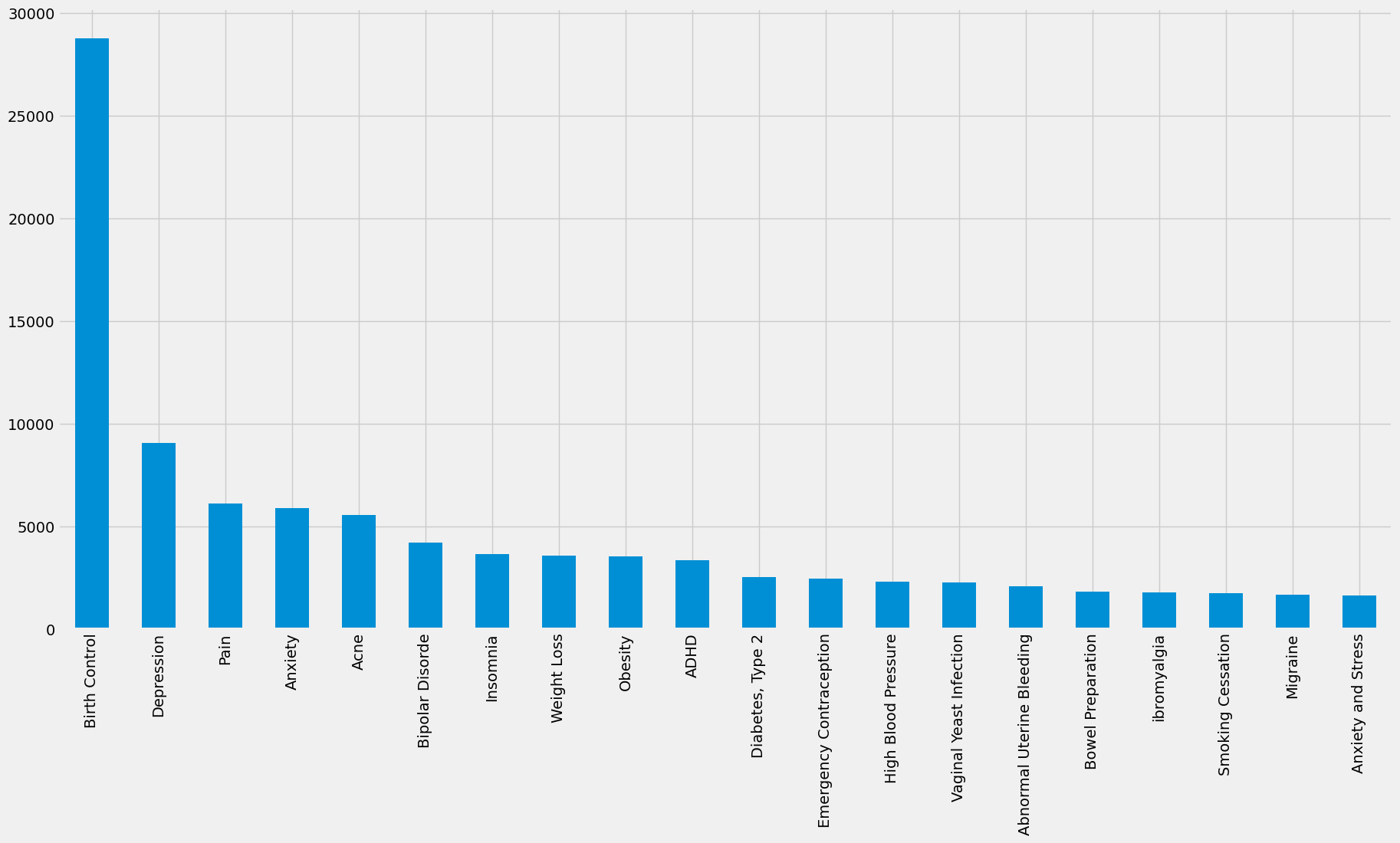
****

There are **884** unique condition presents in data.

**## Top 20 Condition**

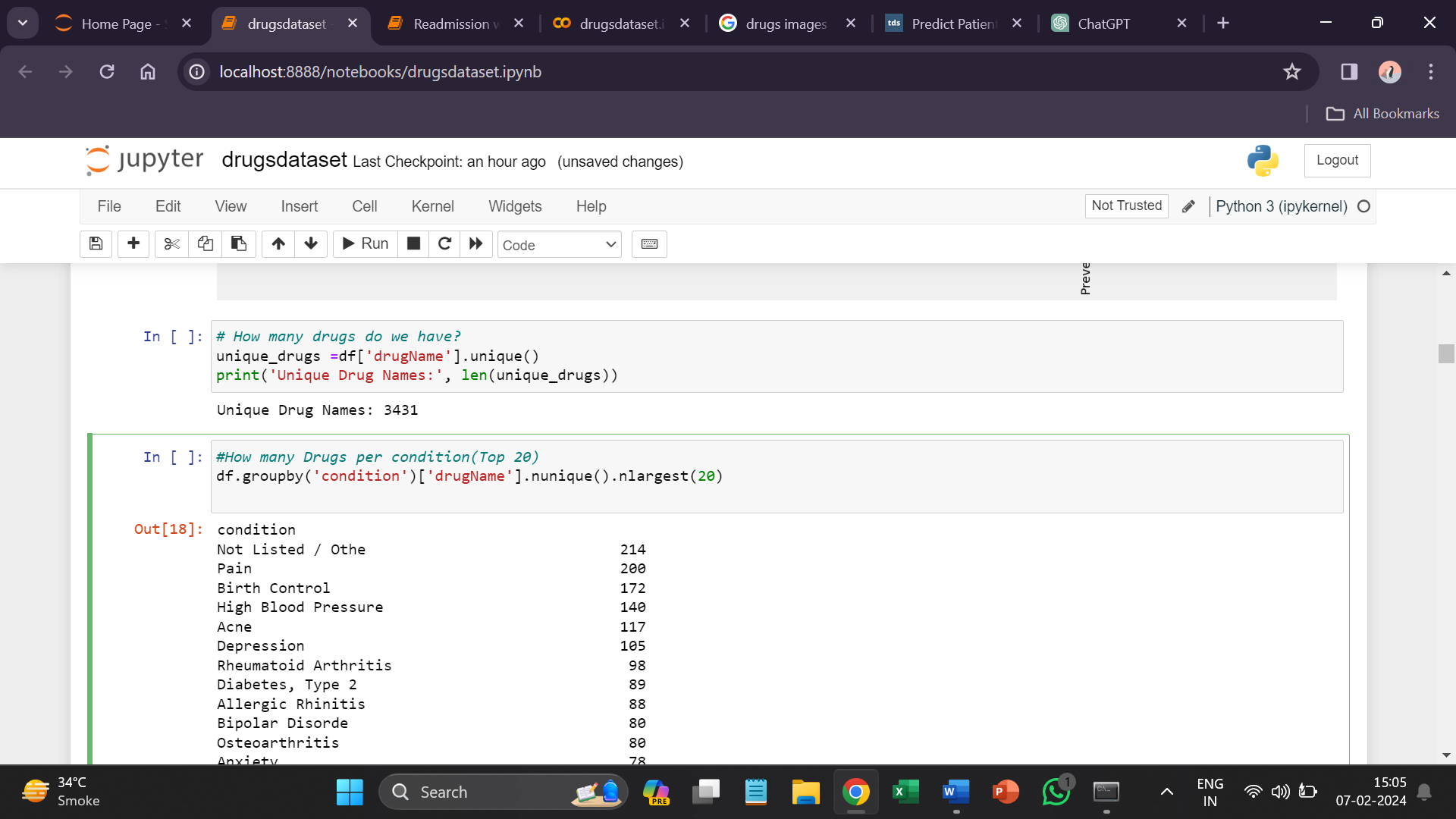


The top 20 condition is followed as Birth Control, Depression, Pain, and Anxiety.



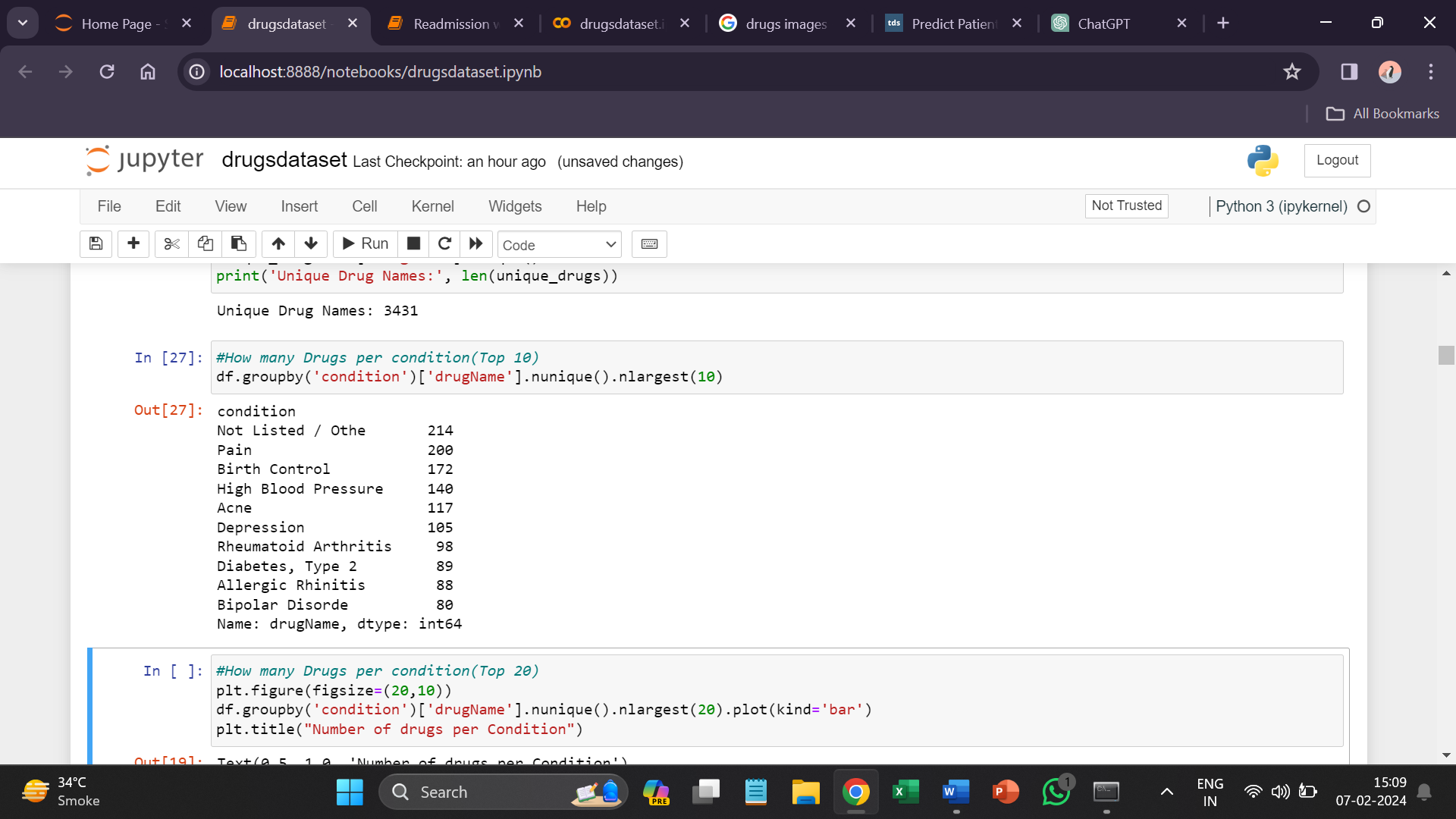
As we can see the most common condition in data is Birth Control and Depression.

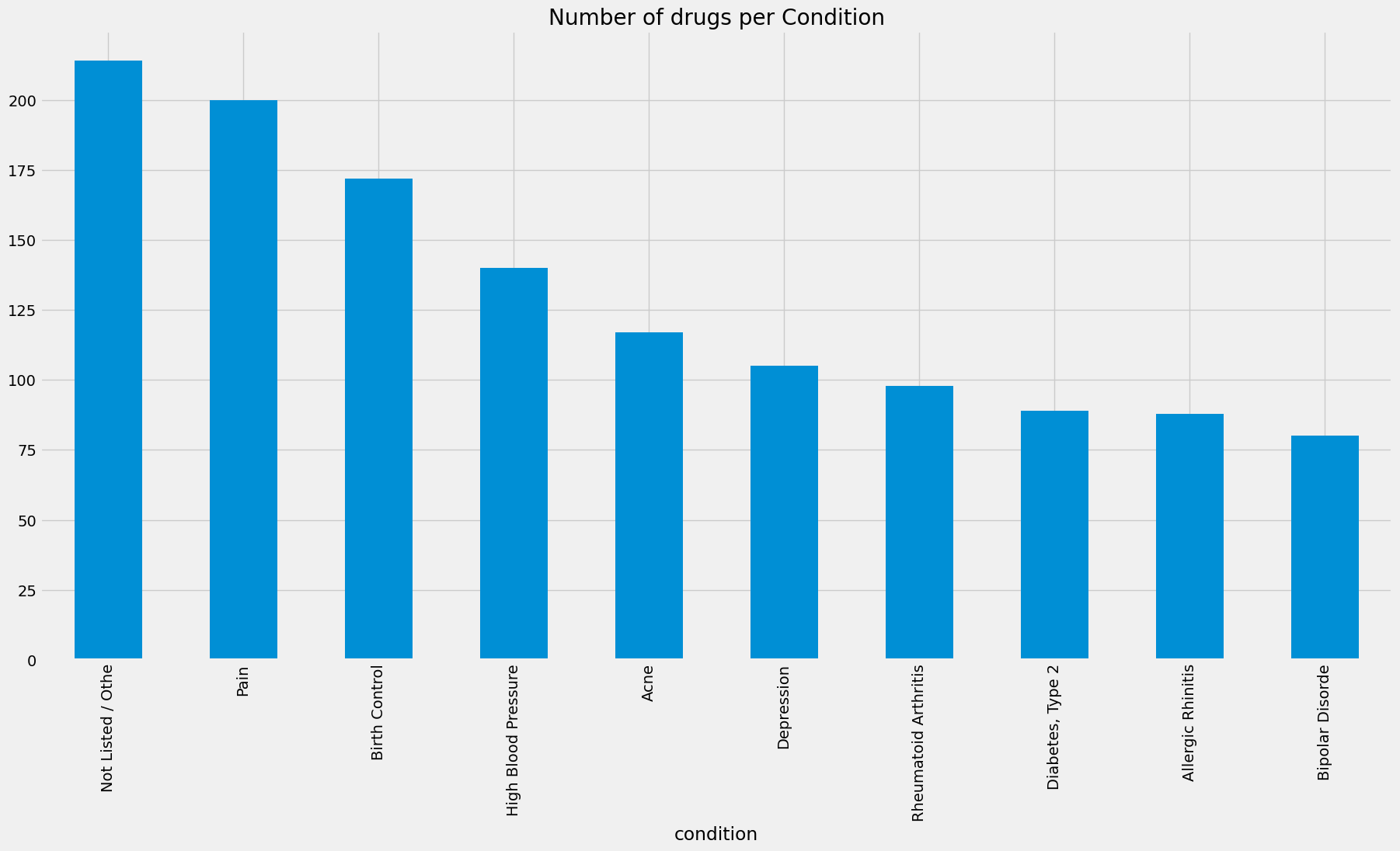
**## How many drugs do we have?**



There are only **3431** unique drugs present.

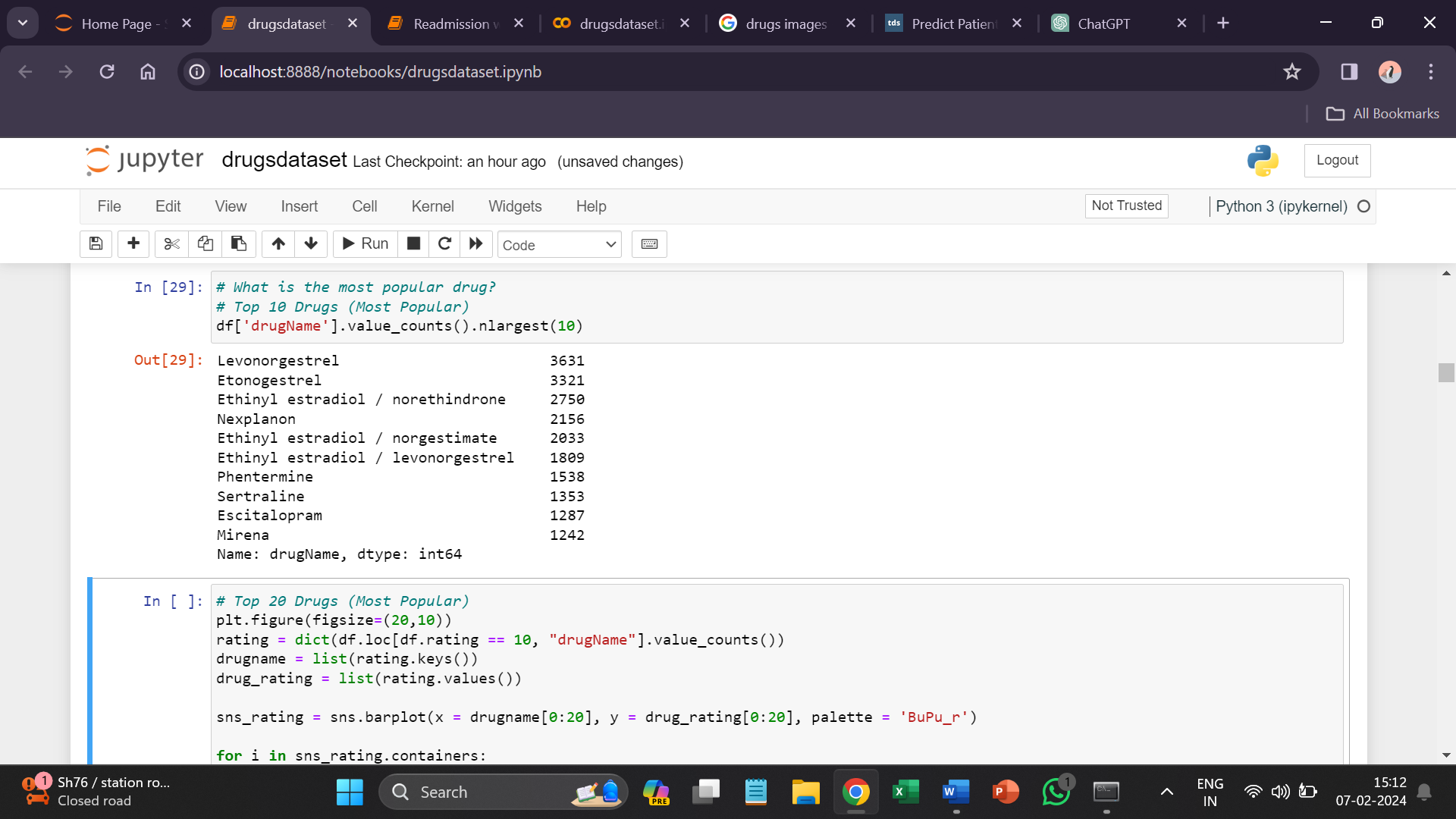
**## How many Drugs per condition**

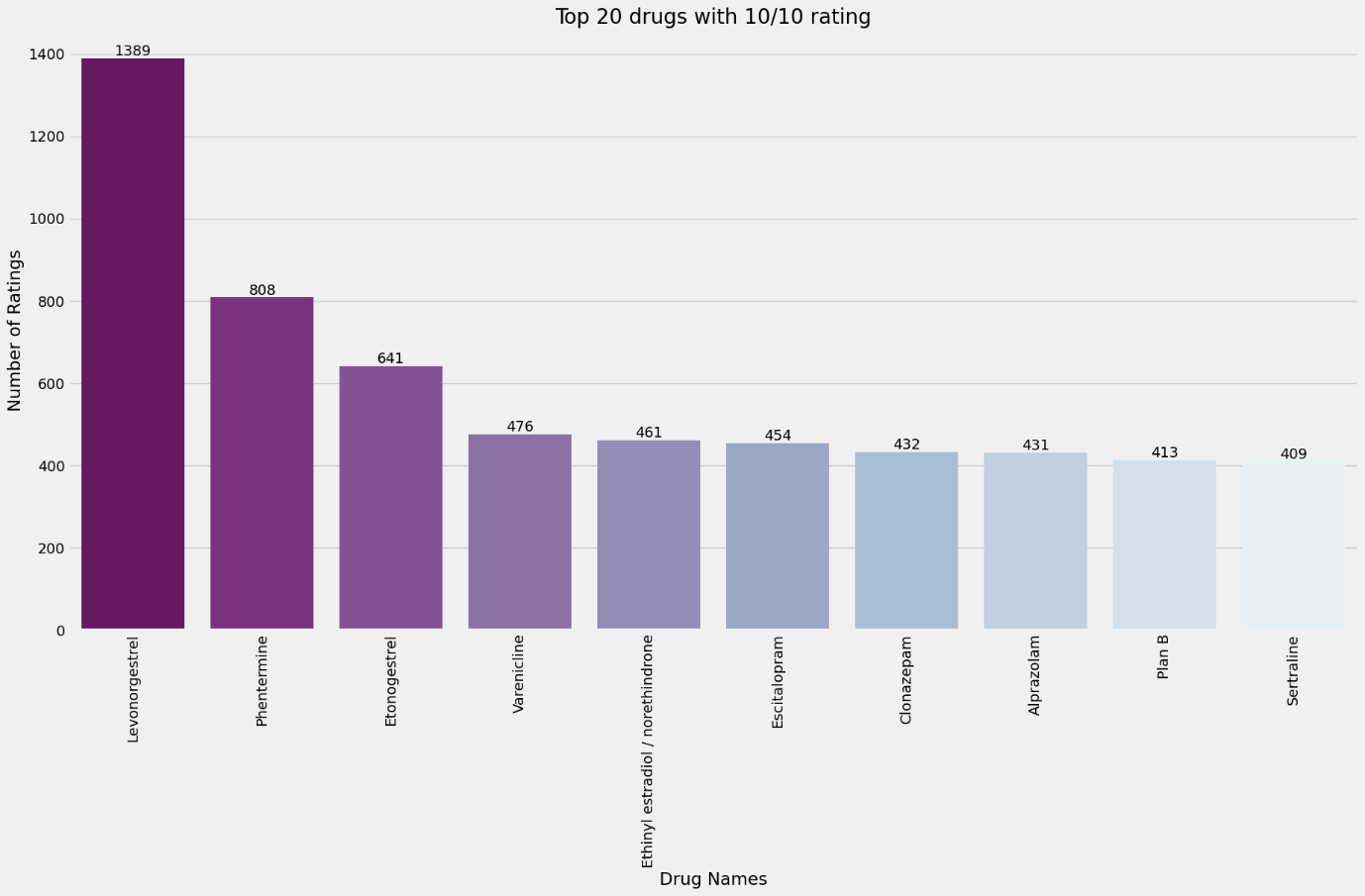




Pain, Birth Control and HBP have the highest number of different drugs for their condition.

**## What is the most popular drug?**

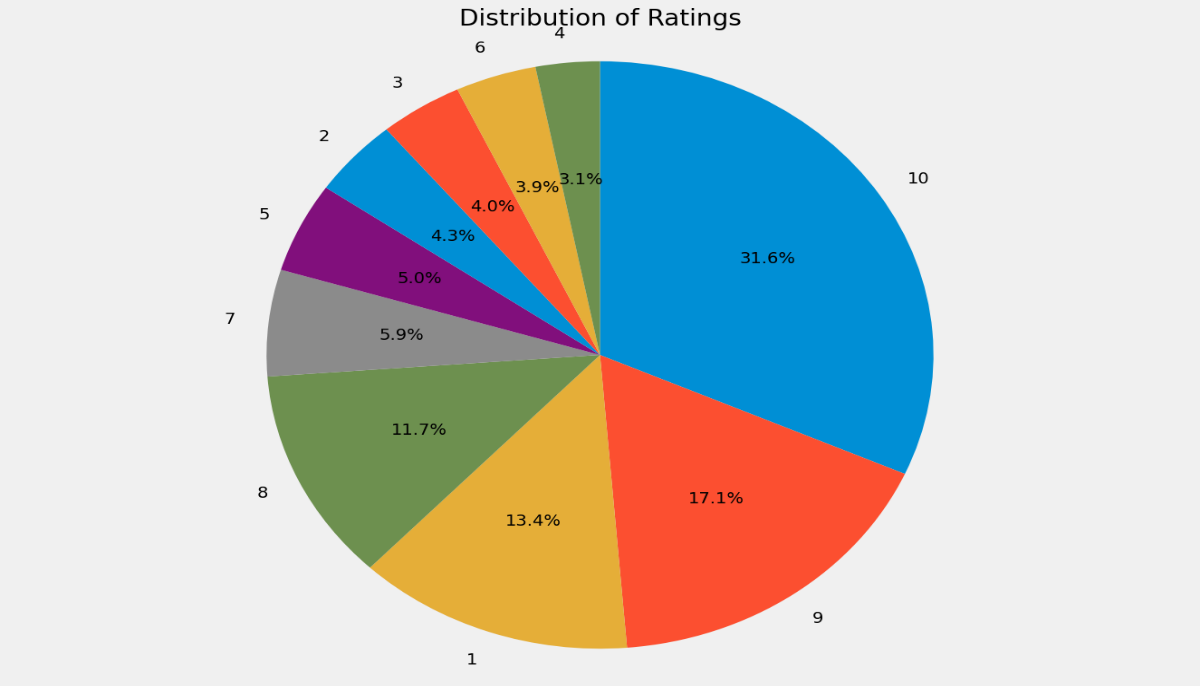




Levonorgestrel, Phentermine, Etonogestrel are the drugs with the highest ratings.

**## Distribution of Rating by size**

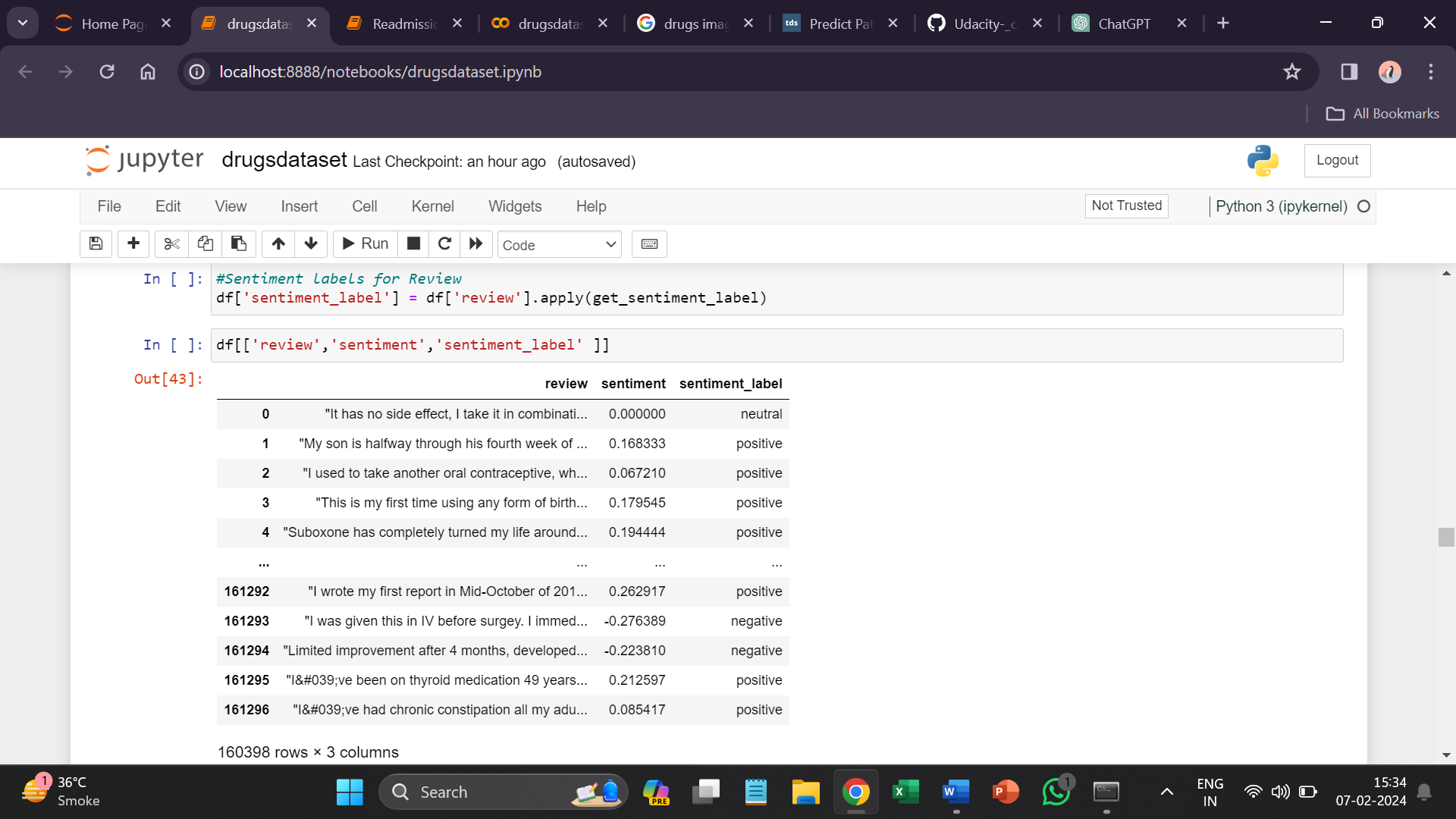


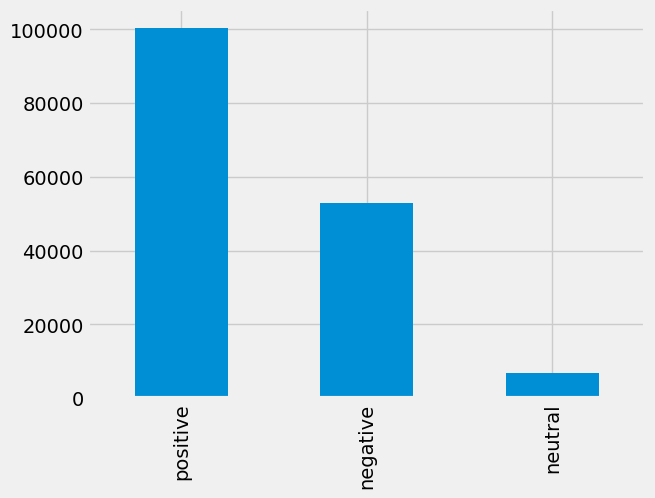


From above we can see that many people have given a rating of 10(31.6%) then 9(17.1%).

**Sentiment Analysis**

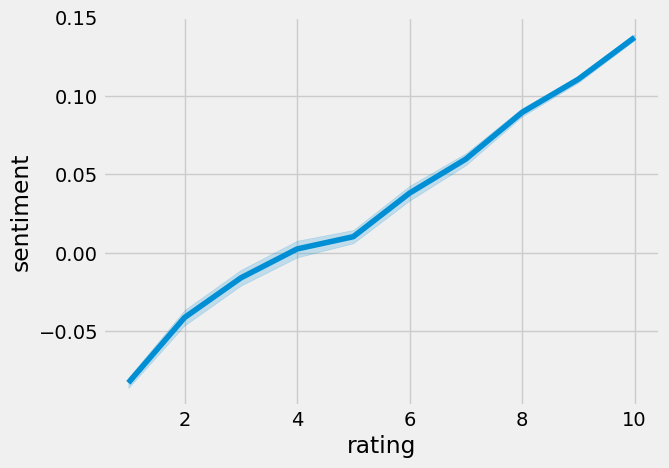
By understanding patient sentiment, healthcare providers and companies can better address patient concerns and improve communication strategies.





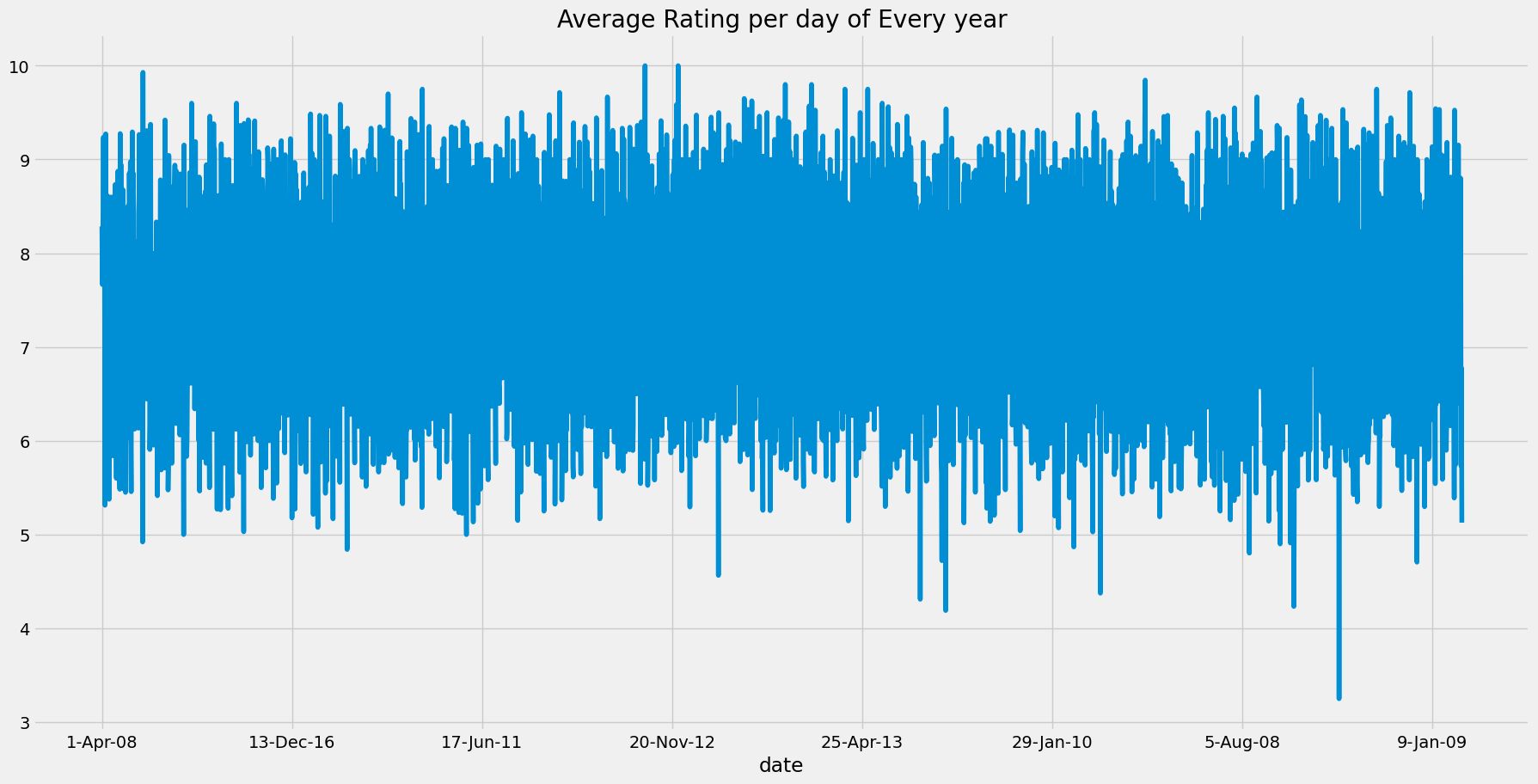
From above we can understand that many people have given positive review to drugs.

**## Correlation between Sentiment and Ratings**



We can see there is **positive correlation** between sentiment and rating which indicates the patient satisfaction based on review.

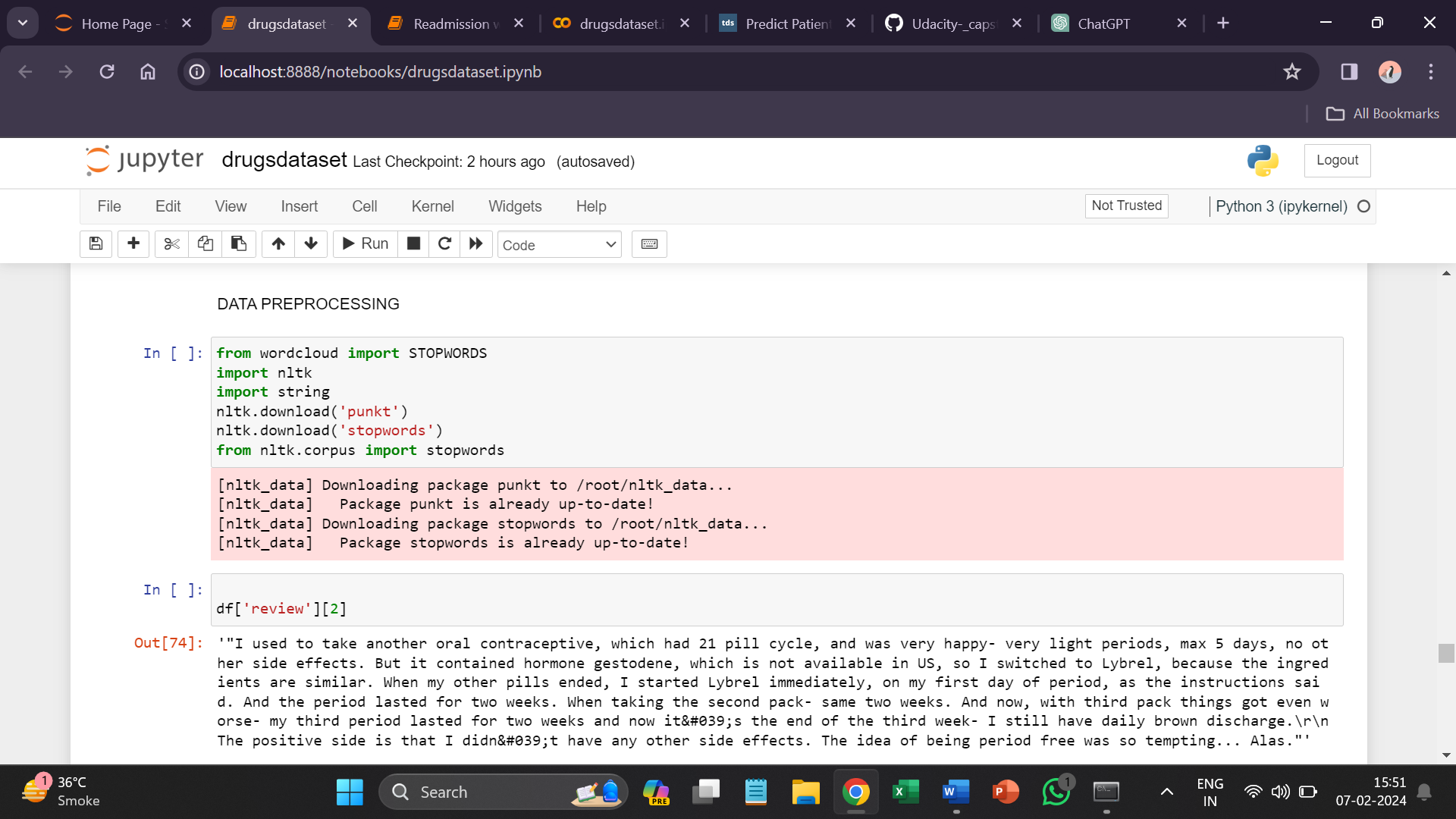
**## Average Rating per day of Every year**



**Preprocessing Reviews**

To make the reviews ready to be fed into any Machine Learning Model, we must preprocess them so that the unnecessary information is removed. We will remove **stop words** and then perform the **stemming** of the words in the corpus.

**Importing all the necessary libraries**



**from wordcloud import STOPWORDS**: This line imports the STOPWORDS set from the wordcloud library. STOPWORDS is a predefined set of common words that are often excluded from text analysis, as they don't typically carry much meaningful information.

**import nltk**: This line imports the Natural Language Toolkit (nltk) library, which is a powerful library for working with human language data.

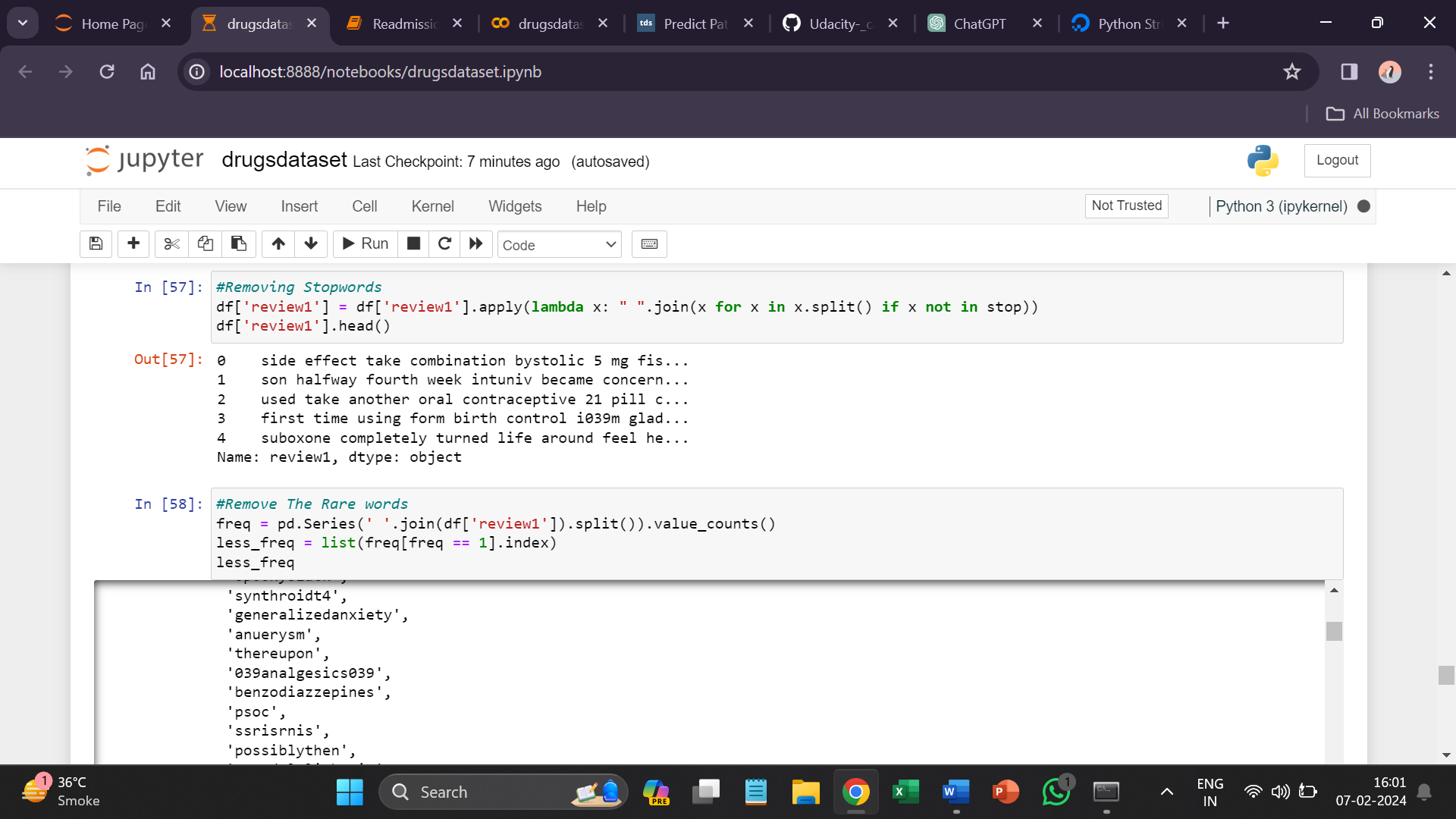
**import string**: This line imports the string module, which provides a collection of string constants, including ASCII characters and various string-related functions.

**nltk.download('punkt')**: Downloads the necessary data for the nltk library related to tokenization. Tokenization is the process of breaking text into individual words or tokens.

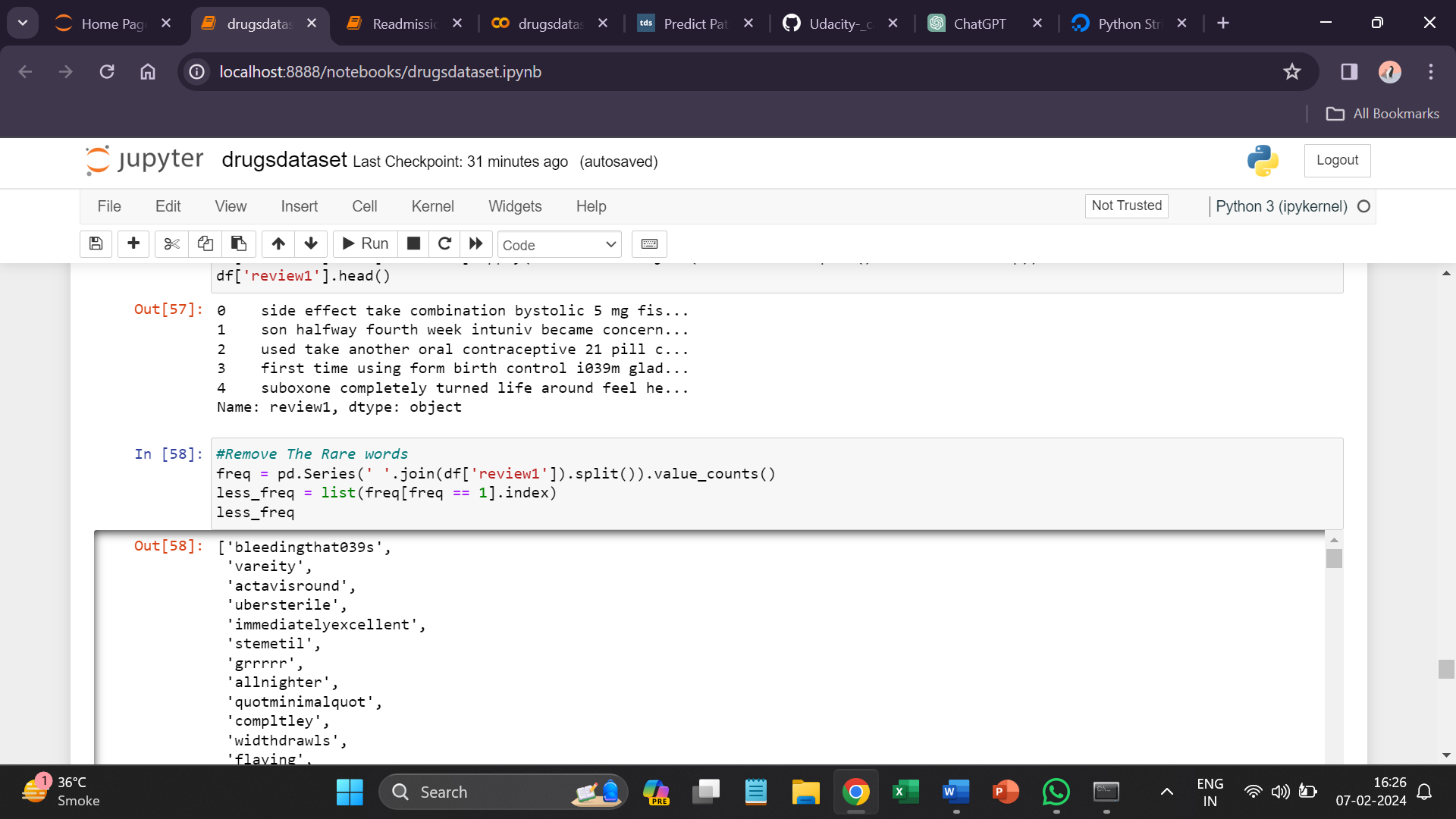
**nltk.download('stopwords')**: Downloads the predefined stopwords from nltk. Stopwords are common words (e.g., "the", "and", "is") that are often removed from text data during processing because they are not informative for many natural language processing tasks.

**from nltk.corpus import stopwords**: This line imports the stopwords from the nltk library. After downloading, you can use these stopwords to filter out common words from your text data.

**Removing Stop words**



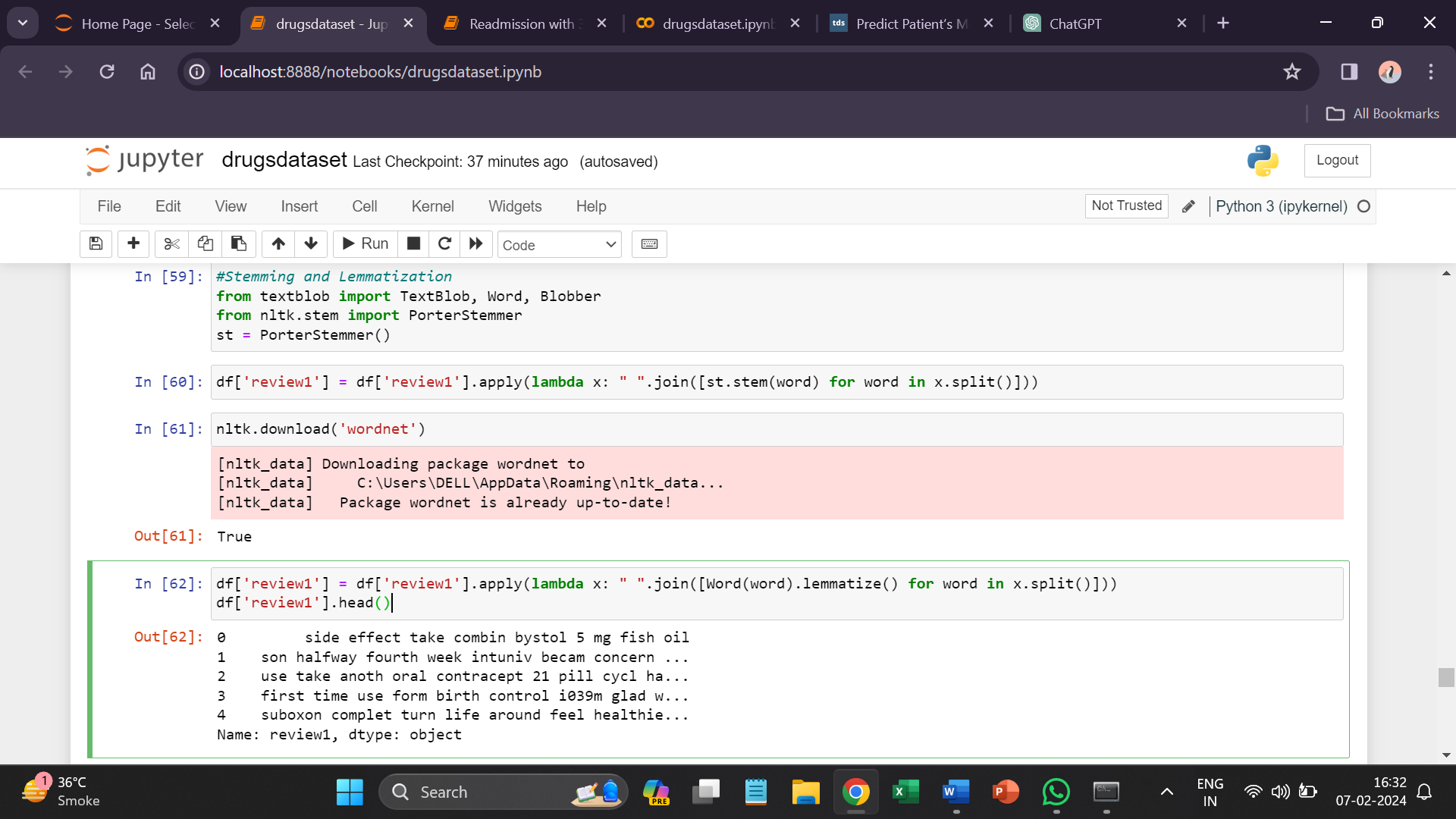
**Removing Rare words**

****

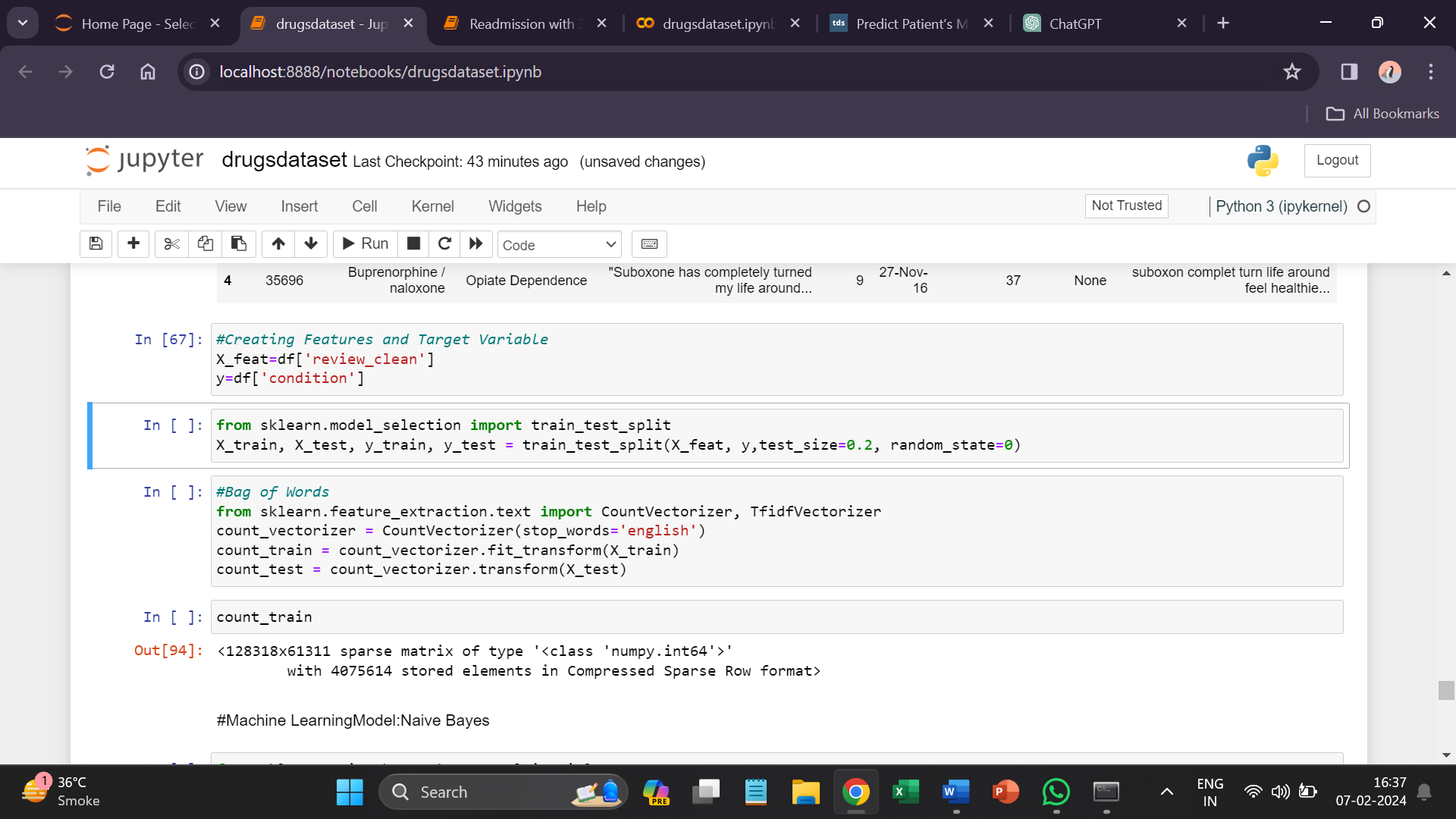
**Stemming and Lemmatization**

Stemming is a text normalization technique that involves reducing words to their base or root form. The Porter stemming algorithm, implemented by Porter Stemmer, is one of the widely used stemming algorithms. It helps in simplifying words to their common base or root, removing suffixes, and is often used to improve text analysis tasks.

Lemmatization is a more sophisticated process that involves reducing words to their base or dictionary form, known as the lemma. Lemmatization considers the context and meaning of words, resulting in valid words.

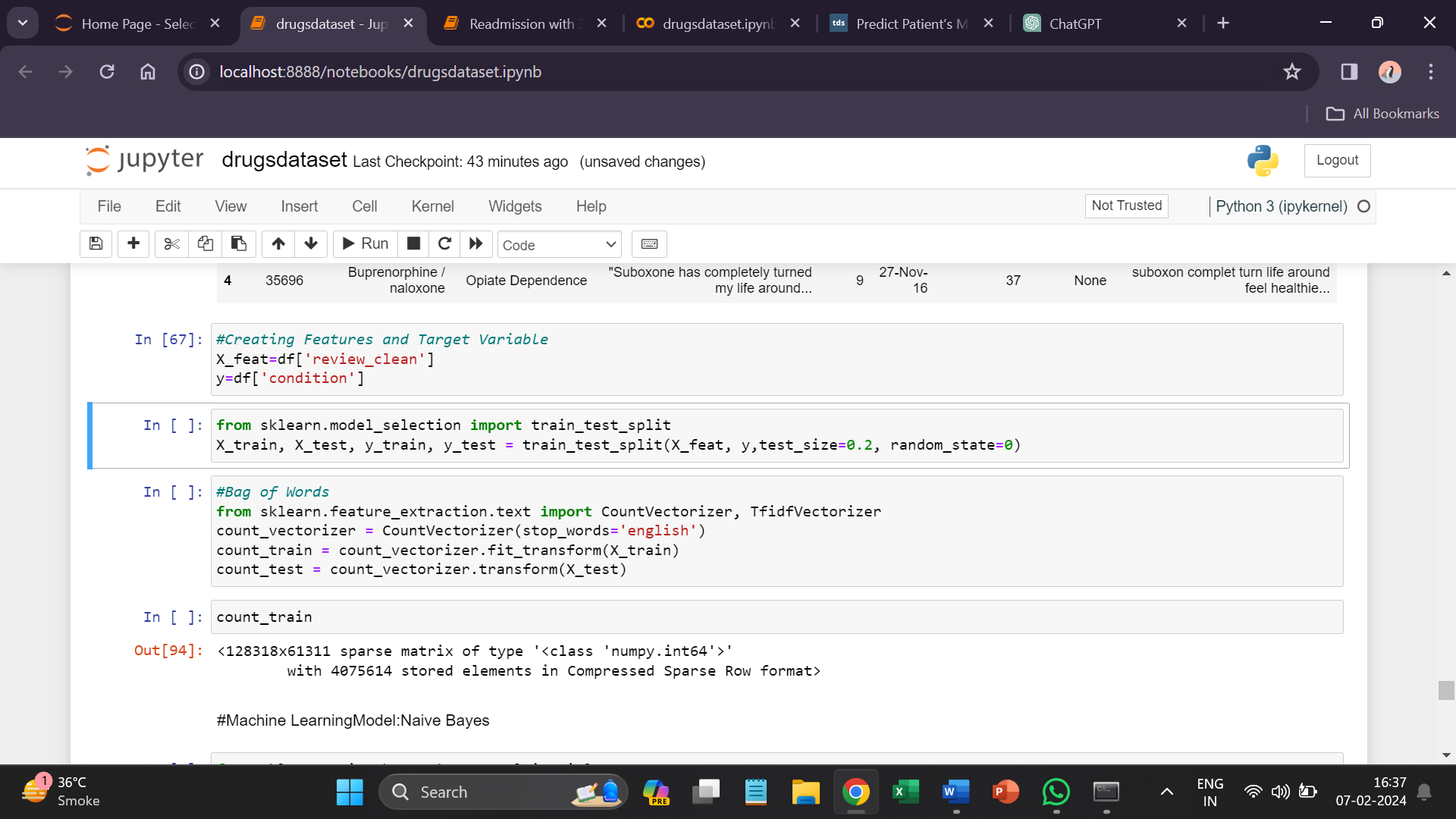


**Splitting the dependent and independent variable**



**Bag of Words**

The Bag of Words (BoW) model is a common technique in natural language processing (NLP) and information retrieval. It represents a document as an unordered set of words, disregarding grammar and word order but keeping track of the frequency of each word.



* It imports the necessary classes from scikit-learn for text vectorization. **CountVectorizer** and **TfidfVectorizer** are both methods for converting text data into numerical vectors.
* **CountVectorizer** is used to convert a collection of text documents to a matrix of token counts (i.e., a numerical representation of the frequency of words in the documents). The stop\_words='english' parameter specifies that common English stop words (like "the," "and," "is") should be excluded from the analysis.

**Creating Machine Learning Model**

1. **Machine Learning Model: Naive Bayes**

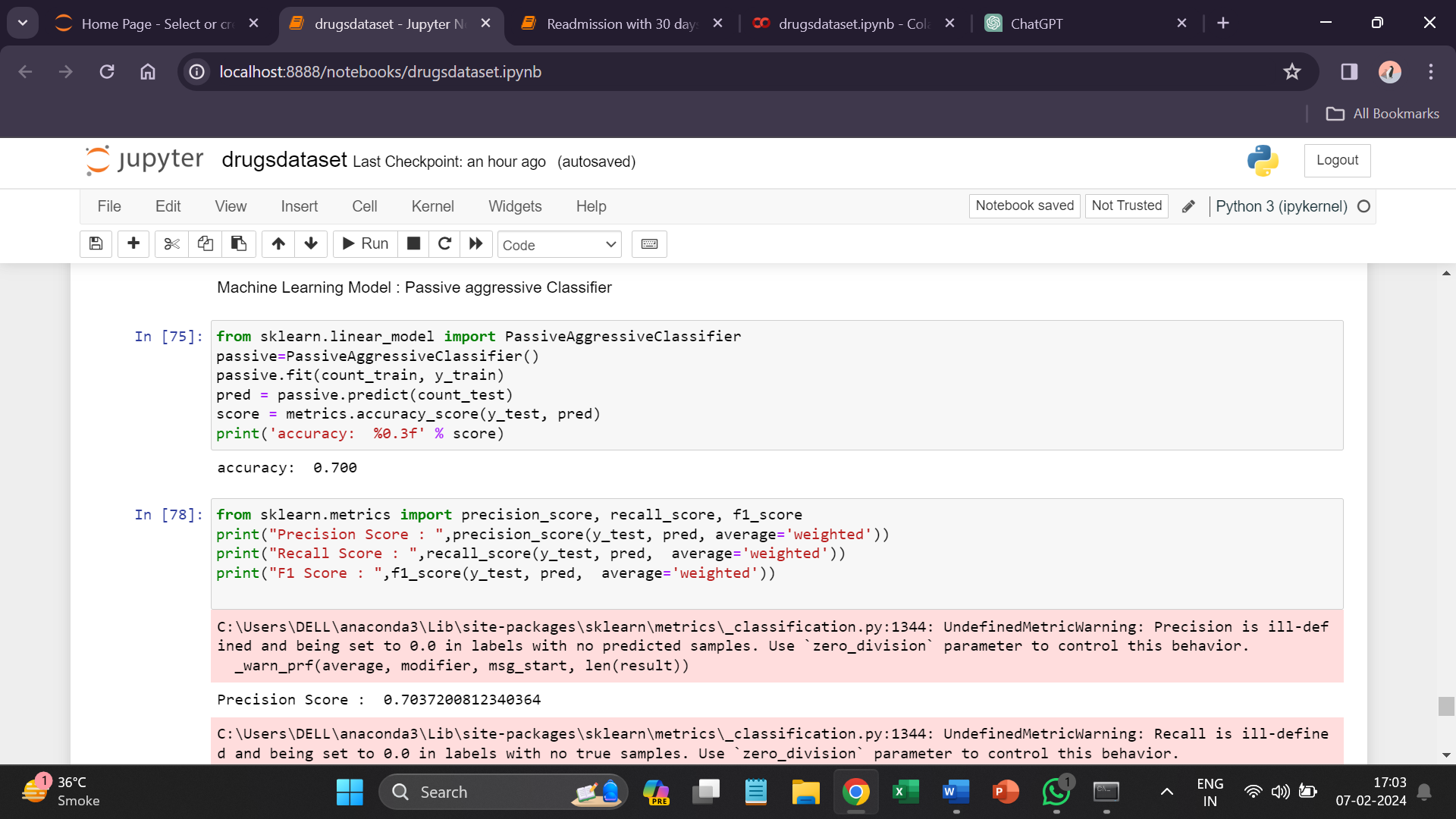
* Multinomial Naive Bayes is a variant of the Naive Bayes algorithm specifically designed for text classification tasks, where the features represent the frequencies of words or other tokens in documents.
* In text classification tasks, documents are typically represented as vectors of word counts or term frequencies. Each document becomes a data point, and each feature represents the frequency of a particular word or term in the document. These feature vectors are then used as input to the Multinomial Naive Bayes classifier.

<<

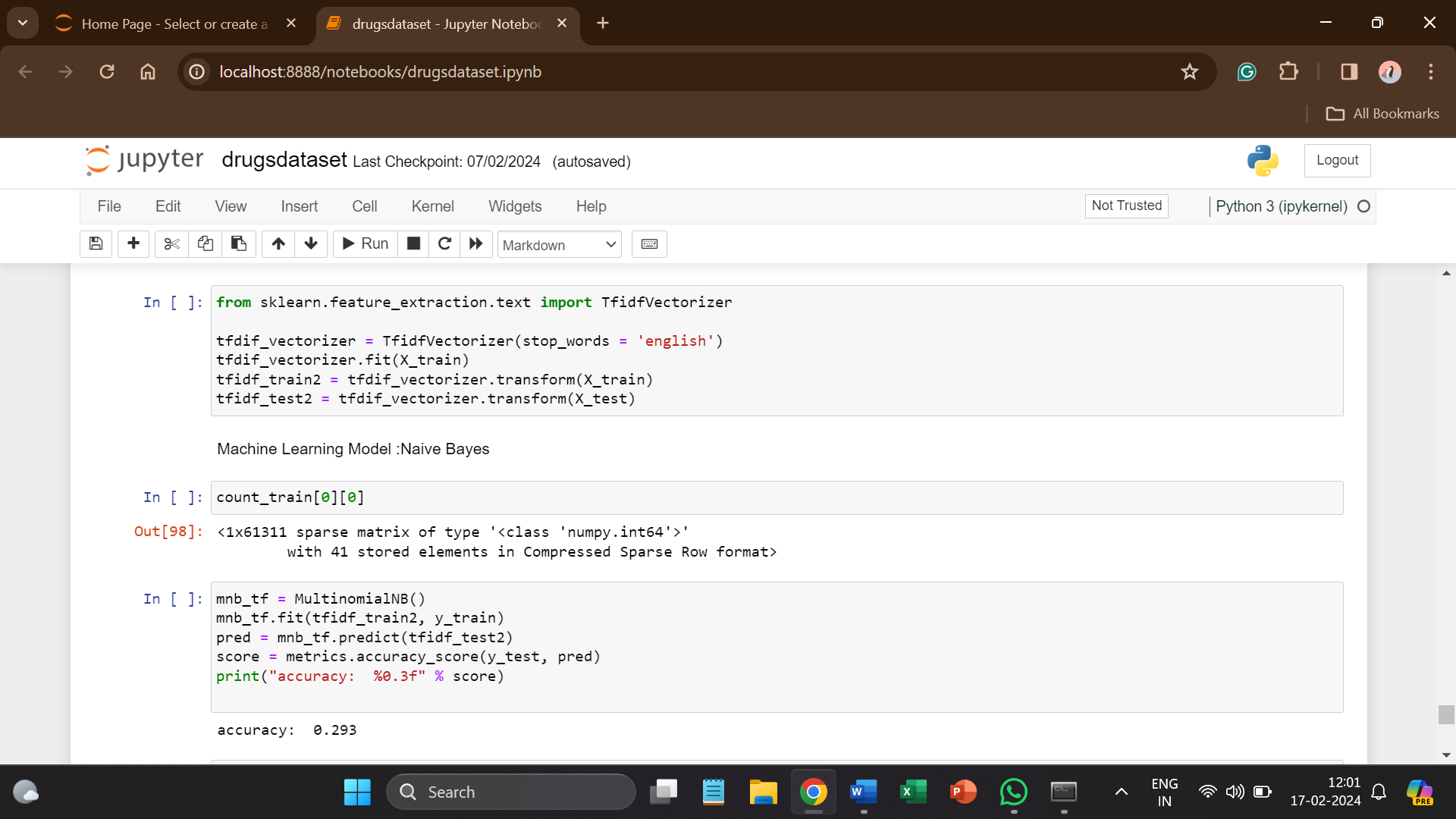


1. **Passive Aggressive Classifier**

* The Passive-Aggressive (PA) algorithm is a machine learning algorithm used for classification tasks, particularly in scenarios where data streams in continuously or where computational resources are limited.
* It's commonly used in real-time applications where quick adaptation to changing data distributions is essential.



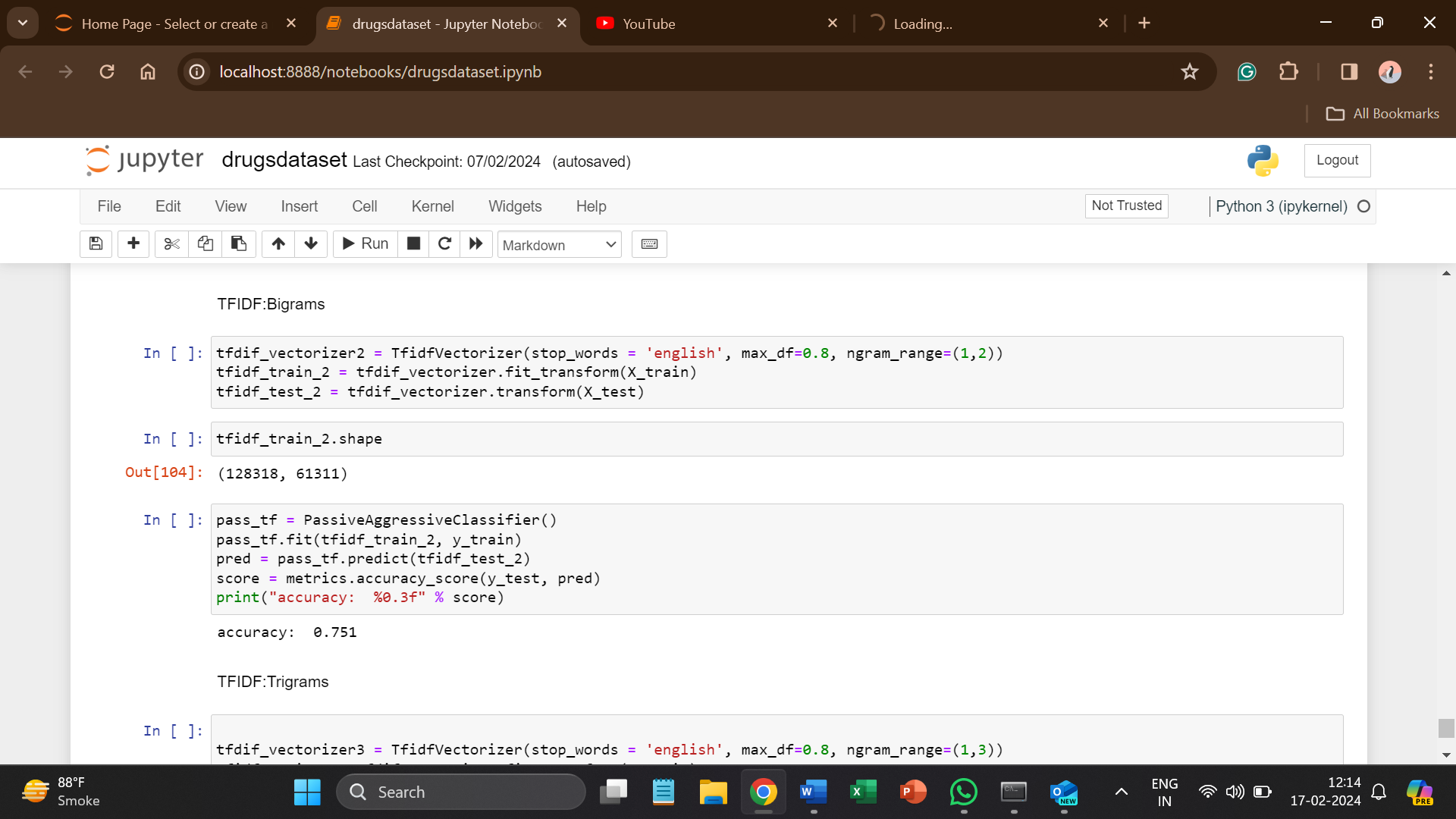
1. **TF-IDF Vectorizer**



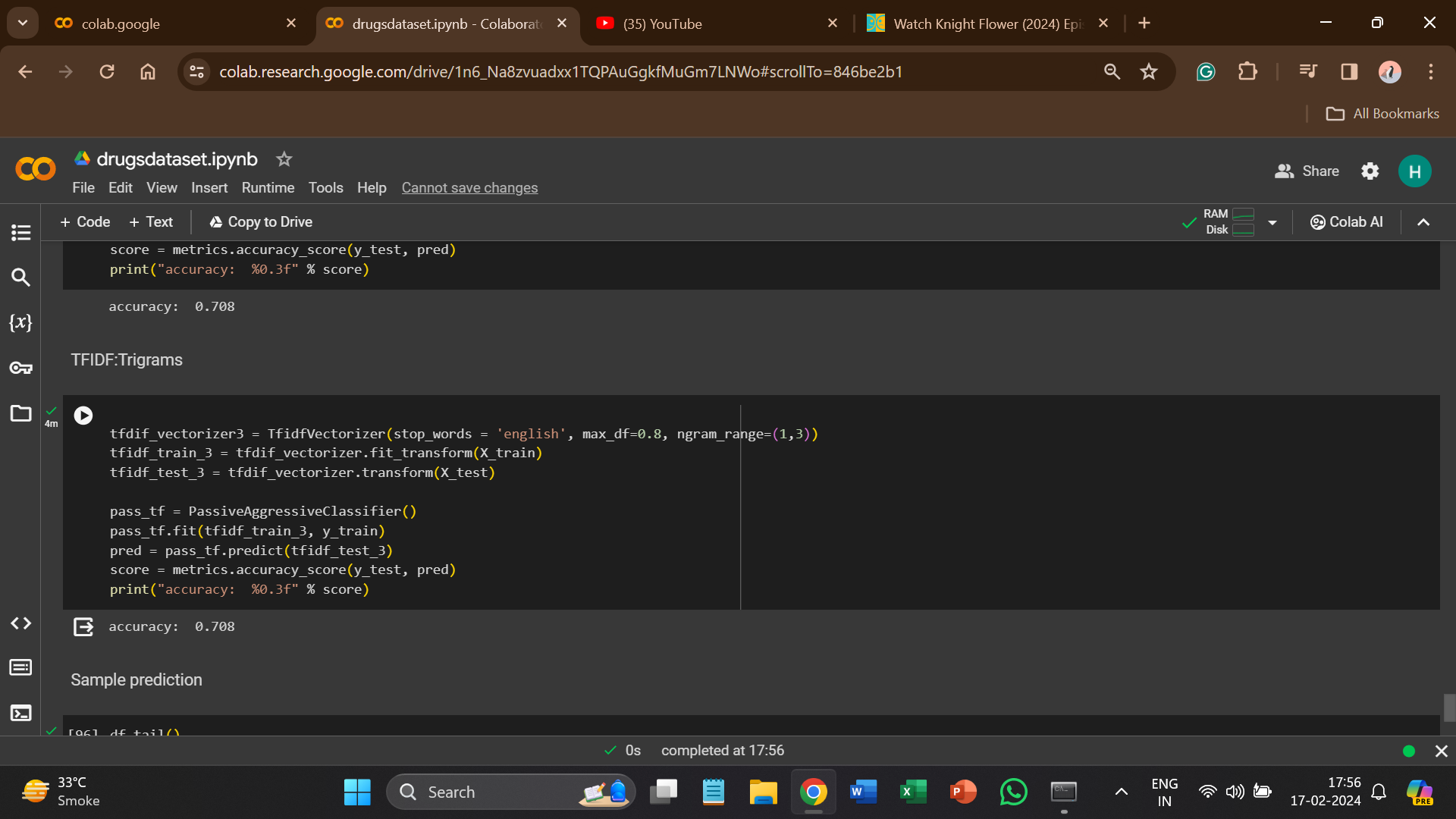
* TF-IDF is a numerical statistic that reflects the importance of a word in a document relative to a collection of documents. It combines two components Term Frequency (TF) and Inverse Document Frequency (IDF).

1. **TF-IDF Bigrams**

In the context of TF-IDF with bigrams. The term frequency considers individual words and adjacent pairs of words in a document. This helps capture the contextual information present in sequences of two words.



1. **TF-IDF Trigrams**

****

* A trigram is a sequence of three consecutive elements, typically applied to words in the context of language processing.
* In the case of text, a trigram refers to a set of three consecutive words that appear together in a sentence or document.