|  |  |
| --- | --- |
| Internship Project Title | Himanshu |
| Name of the Company | TCS iON |
| Name of the Industry Mentor | Himdweep Walia |
| Name of the Institute | Amity University Online |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Start Date | End Date | Total Effort (hrs.) | Project Environment | Tools used |
| 20-04-2024 | 07-06-2024 | 210 Hours | Python Virtual Internship Project | Jupyter Notebook, Google Collab, TensorFlow. keras, matplotlib and Google Search for learn about drugs effect, Kaggle |

**TABLE OF CONTENTS**

[ACKNOWLEDGEMENT 3](#_Toc142647963)

[OBJECTIVE 3](#_Toc142647964)

[INTRODUCTION / DESCRIPTION OF INTERNSHIP 4](#_Toc142647965)

[INTERNSHIP ACTIVITIES 5](#_Toc142647966)

APPROACH / METHODOLOGY  [7](#_Toc142647967)

[DATASET INFORMATION 7](#_Toc142647968)

[EXPLORATORY DATA ANALYSIS 9](#_Toc142647969)

[VISUALIZATION 12](#_Toc142647971)

[PREPROCESSING 18](#_Toc142647971)

[ENCODING 20](#_Toc142647972)

[FEATURE SELECTION 20](#_Toc142647972)

[ALGORITHMS 22](#_Toc142647973)

[RANDOM FOREST CLASSIFIER 22](#_Toc142647974)

[K-NEAREST NEIGHBORS CLASSIFIER 22](#_Toc142647975)

[GRADIENT BOOSTING CLASSIFIER 23](#_Toc142647976)

[LOGISTIC REGRESSION CLASSIFIER 23](#_Toc142647977)

[HYPERPARAMETER TUNING 24](#_Toc142647978)

[CHALLENGES & OPPORTUNITIES 24](#_Toc142647979)

[REFLECTIONS ON THE INTERNSHIP 25](#_Toc142647980)

[OUTCOME / CONCLUSION 25](#_Toc142647981)

[ENHANCEMENT SCOPE 26](#_Toc142647982)

[LINK TO CODE AND EXECUTABLE FILE 26](#_Toc142647983)

## **ACKNOWLEDGEMENT**

I Himanshu would like to express my gratitude for the opportunity to undertake the internship on the topic "Classification Model-Build a Model that Classifies the Side Effects of a Drug" as a part of the course completion requirements for MCA Machine Learning at Amity University Online.

I extend my sincere appreciation to TCS iON for providing me with the platform to apply theoretical knowledge gained during my academic studies to real-world scenarios. The hands-on experience gained through this internship has been invaluable in enhancing my understanding of machine learning concepts and their practical applications.

I am also thankful to the faculty members at Amity University Online & TCS iON for their guidance and support throughout the duration of the internship. Their expertise and mentorship have played a significant role in shaping my learning journey and preparing me for future endeavors in the field of machine learning.

Furthermore, I would like to acknowledge the contributions of my Industry Mentor Himdweep Walia at TCS iON for their assistance during the internship period. Their insights and feedback have been instrumental in refining my skills and achieving the objectives of the project.

In conclusion, I am grateful for the enriching experience provided by both Amity University Online and TCS iON, and I look forward to applying the knowledge and skills acquired during this internship in my future academic and professional pursuits.

## **OBJECTIVE**

The primary objective of this project is to create a robust classification model capable of effectively predicting and categorizing the potential side effects associated with a particular drug. By harnessing patient attributes such as age, gender, and race, the model aims to provide accurate insights into the likelihood of various side effects. The ultimate goal is to assist healthcare professionals in better understanding the potential risks associated with specific medications, thereby enabling informed decision-making and personalized treatment strategies.

## **INTRODUCTION / DESCRIPTION OF INTERNSHIP**

The field of healthcare and pharmaceuticals has always been at the forefront of technological advancements, seeking innovative ways to ensure patient well-being and safety. In this context, the internship topic of "Classification Model: Build a Model that Classifies the Side Effects of a Drug" assumes paramount significance. This internship delves into the realm of data science and machine learning, where the objective is to create a predictive model that can effectively categorize the potential side effects of various drugs.

Unleashing the power of machine learning in health care redefined treatment precision, personalized care, and unlocked new possibilities. With real-time patient data, drug side effects are classified with accuracy, empowering smarter decisions for better outcomes. Discover the future of healthcare, where innovation meets value-based care.

Side effects are unwanted effects that can occur alongside the desired effects of a drug or medication. They vary based on factors like age, disease, gender, race, and health. Starting, changing, or stopping medication can trigger side effects that lead to non-compliance. Severe cases may require dosage adjustments or additional medication. Lifestyle changes can help minimize side effects. Classifying side effects for each drug is challenging, but machine learning eases the process while maintaining accuracy. Research reveals differences in drug response among racial and ethnic groups, emphasizing the need for inclusive treatment policies that accommodate individual needs.

For this project, a dataset of drug effects lacked ’Name’, ‘Race’, ‘Gender’ and ‘Age’ features. The dataset used for classification includes the drug name, side effects, etc. of different users of drugs. Datasets were combined. Supervised machine learning classifiers are used in building the model and fitting the data into the model.

## **INTERNSHIP ACTIVITIES**

Day-1 Clear assessments, watched all content, learn about workflows

Day-2 Learn about project, go through some reference material and learn about medicine properties effect, advantage and disadvantage.

Day-3 Learn about dataset what will be needed for project, do some more research about drug and research about how can we create dataset by researching some third-party webpages.

Read through available drugs trial dataset:

<https://archive.ics.uci.edu/dataset/461/drug+review+dataset+druglib+com>

Day-4 Drug data which was gathered by academics and compiled in open databases, computational approaches for side effect forecasting will be used to create a model that classifies the trial data of a drug based on their age, gender and race.

1. We will be utilizing a data set consisting of 4,00,000 patients containing the following details for each patient: Name, Age , Gender , Race , Side effects .

Read through available drugs trial dataset : <https://www.fda.gov/drugs/information-consumers-and-patients-drugs/fdas-drug-review-process-ensuring-drugs-are-safe-and-effective>

Day-5 Learn about machine learning, what is the purpose of machine learning., learn about how can we use the ML Algorithms on data, watch videos reference which available on YouTube.

Day-6 Learned about how to gather data and create dataset

Day-7 Learned about what is train, test data

Day -8 Learned about what is regression and classification

Day -9 Learned about data analysis and some basic algorithms

Day -10 Learned about data pre-processing steps for Machine Learning & Data analytics.

Day-11 Learned about Data pre-processing with Python on dummy dataset

Day-12 Learned about data Cleaning on dummy iris dataset in Pandas and NumPy.

Day-13 Learned about data transformation process in Machine Learning.

Day -14 Learned about data transformation process using a live Python use case scenario

Day-15 Learned about Data reduction in data analysis.

Day-16 Learned about Dimensionality reduction in Python

Day-17 Practice training and testing on dummy dataset in python

Day -18 Learned about brief about training and test dataset

Day -19 Learned about data analysis and some basic algorithms

Day -20 Learned about the accuracy checking of Machine learning models

Day-21 Learned about the accuracy checking of Machine learning models

Day-22 Learned about Classifiers and their types in Machine learning.

Day-23 Learned about Perceptron model in Machine Learning.

Day -24 Learned about Logistic Regression in machine learning

Day-25 Learned about Naive Bayes Algorithm in Machine Learning.

Day-26 Learned about K-Nearest Neighbor(KNN) Algorithm for Machine Learning:

Day-27 Learned about Support Vector Machine (SVM) Algorithm Works In Machine Learning.

Day-28 Learned about Random Forest classifier in Machine learning.

Day-29 Implement a simple machine learning algorithm in Python using Scikit-learn

Day-30 Learned about Creating Classification Report and Confusion Matrix

Day-31 Standardization

Day-32 Learn about the visualization libraries Matplotlib

Day-33 Learn about the data modelling and model fitting

Day-34 Learn about the data Underfitting and Overfitting.

Day-35 Learn Fitting the data to model using python

Day-36 Project Hands-on - Getting drugs dataset and analysing the dataset

Day-37 Project Hands-on - Import related modules and dataset file

Day-38 Project Hands-on – Explore the data By following Exploratory Data Analysis

Day-39 Project Hands-on – Visualization the data By different type of charts

Day-40 Project Hands-on – Preprocessing the data

Day-41 Project Hands-on – Encoding the data

Day-42 Project Hands-on – Feature Selection

Day-43 Project Hands-on – Model Building

Day-44 Project Hands-on – Hyper Parameter Tuning

Day-45 Project Hands-on – Conclusion

**Approach / Methodology**

**Data Set Information**

The provided dataset comprises essential information regarding patients and their interactions with different drugs. It is structured with distinct features that capture various aspects of these interactions, offering insights into their potential effects and outcomes.

**Attribute Information:**

**NAME OF FEATURE DESCRIPTION**

PATIENT NAME Name of patients

GENDER Gender of the patients

AGE Age of the patients

DRUG NAME Name of various drugs

RACE Race of patients

SIDE EFFECTS Column which contains all the side effects of drugs.

The dataset includes the six features shown in the table above and 400000 records. This dataset is valuable for conducting analyses related to drug interactions, gender-based responses, racial disparities, and the prevalence of side effects associated with different drugs. Researchers and analysts can utilize this dataset to draw meaningful conclusions about the relationships between these factors and make informed decisions in the fields of healthcare and pharmaceuticals.

**PATIENT NAME:**

This Column having Name of patient who takes the drug.

**GENDER:**

This Column contain Gender of the patient.

**AGE:**

This Column contain Age of the patient. The age of the all patient between 14 to 90 years.

**DRUG NAME:**

In naming drugs, the most important considerations are avoiding drug names that are too similar to existing names—and therefore might compromise patient safety—and making sure the drug name communicates accurate information about the action or use of the substance.

The term "generic name" has several meanings as regards drugs:

* The chemical name of a drug.
* A term referring to the chemical makeup of a drug rather than to the advertised brand name under which the drug is sold.
* A term referring to any drug marketed under its chemical name without advertising.

"Diazepam" is an example of the chemical (generic) name of a sedative. It is marketed by some companies under its generic name and by other companies under brand names such as Valium or Vazepam.

**RACE:**

A concept used to describe a group of people who share physical characteristics, such as skin color and facial features. They may also share similar social or cultural identities and ancestral backgrounds. There are many racial groups, and a person may belong to or identify with more than one group

**SIDE EFFECTS:**

* Start taking a new drug or dietary supplement (for example, vitamins)
* Stop taking a drug that you’ve been on for a while, or
* Increase or decrease the dose (amount) of a drug that you take.

Common side effects include upset stomach, dry mouth, and drowsiness. A side effect is considered serious if the result is: death; life-threatening; hospitalization; disability or permanent damage; or exposure prior to conception or during pregnancy caused birth defect.

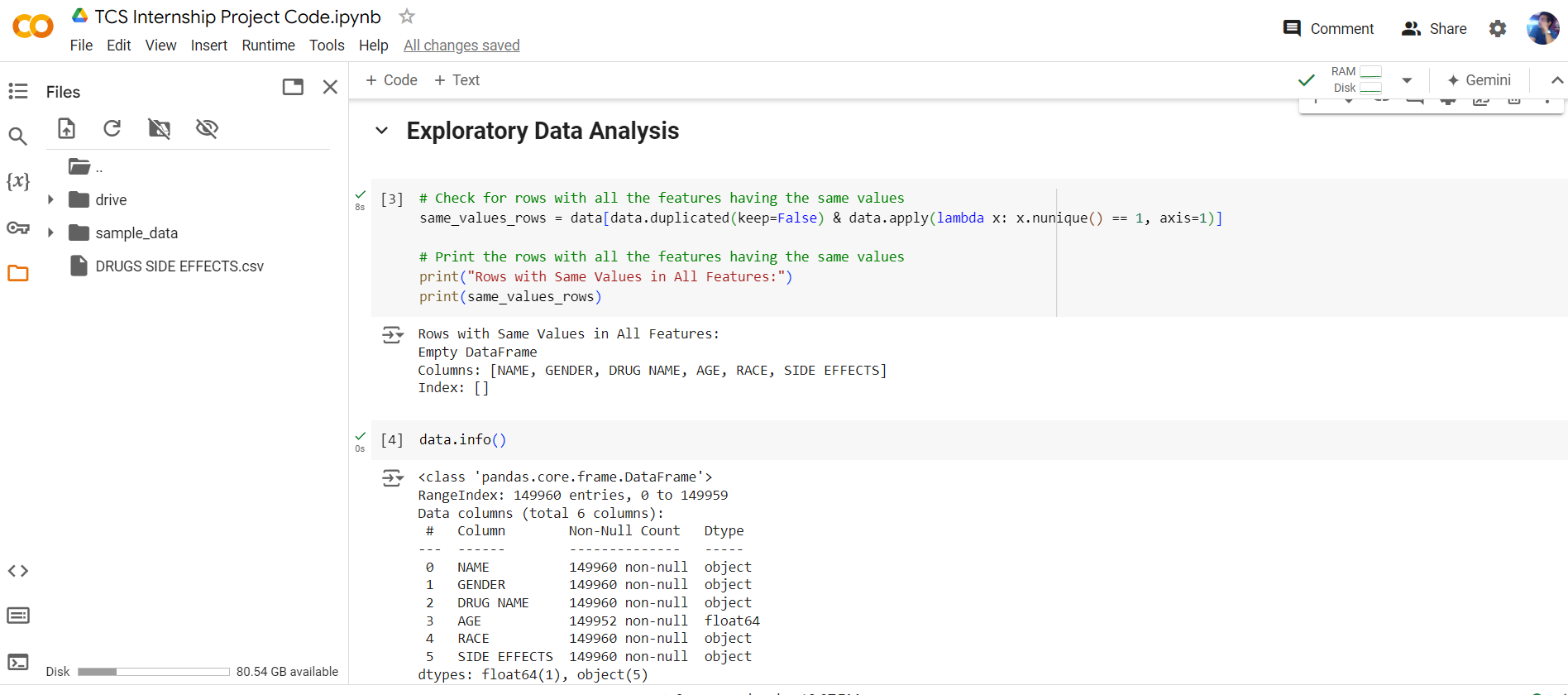
* Moderate Side Effects
* Severe Side Effects
* Extremely Severe Side Effects
* Mild Side Effects
* No Side Effects

**Dataset Link :-** [**https://github.com/Himanshu-2012/TCS-iON-RIO-210-Internship-Build-a-Classification-Model-for-Drug-Trials-Dataset**](https://github.com/Himanshu-2012/TCS-iON-RIO-210-Internship-Build-a-Classification-Model-for-Drug-Trials-Dataset)

I Have uploaded the dataset and all other report in my personal Github account

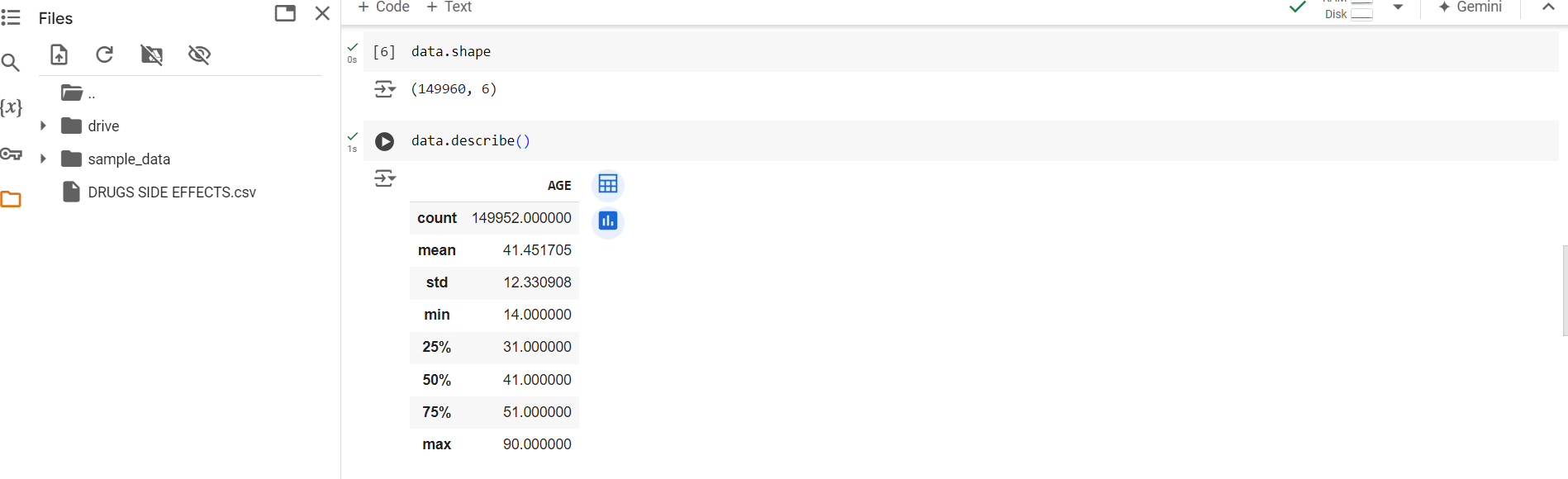
**Exploratory Data Analysis**

First, we explore the data, what is the data type of the columns check the missing values and unique count, check the Type of gender, Type of side effects. Describe the data this can show us the all statistics value of the data.



data.describe()

The describe() method returns description of the data in the DataFrame. If the DataFrame contains numerical data, the description contains these information for each column: count - The number of not-empty values. mean - The average (mean) value



**Check the missing values and unique values in the data by the isna(), nuique().**

The data.isna().sum() function in pandas is used to count the number of missing (NaN) values in each column of a DataFrame. Here is a step-by-step breakdown of what it does:

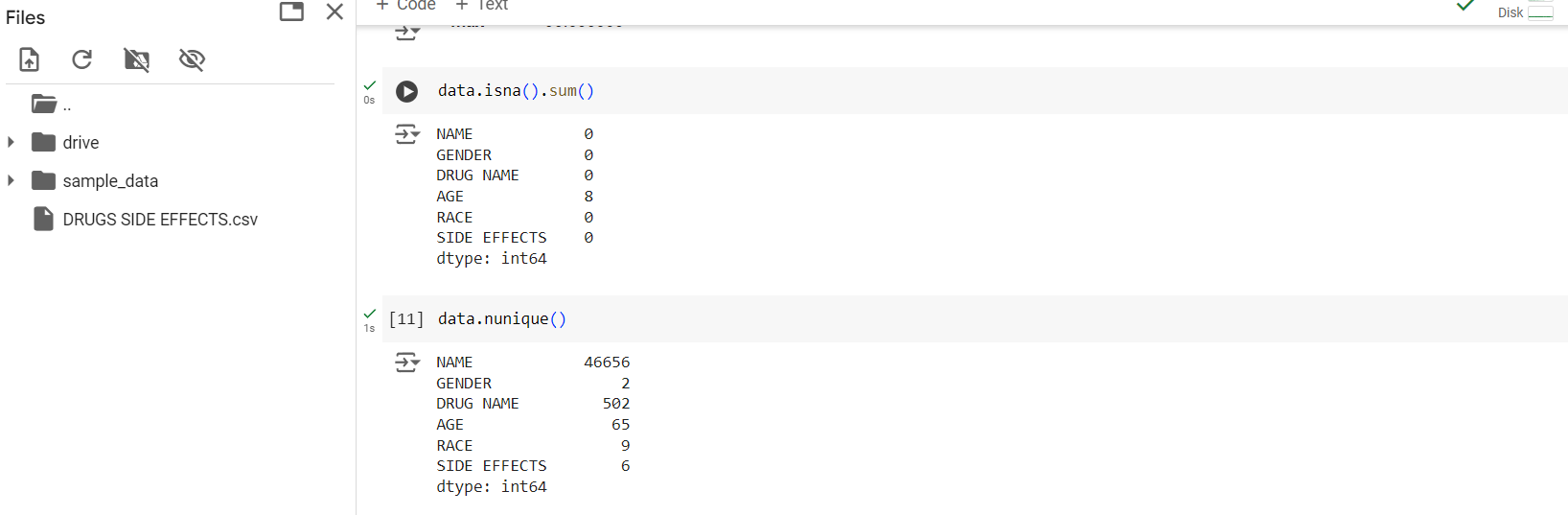
data is your pandas DataFrame.

isna() is a method that returns a DataFrame of the same shape as data, where each element is True if that element is NaN (missing), and False otherwise.

sum() is then called on this DataFrame of boolean values. When sum() is applied to boolean values, True is treated as 1 and False as 0. Therefore, summing along each column gives the total count of NaN values in that column.

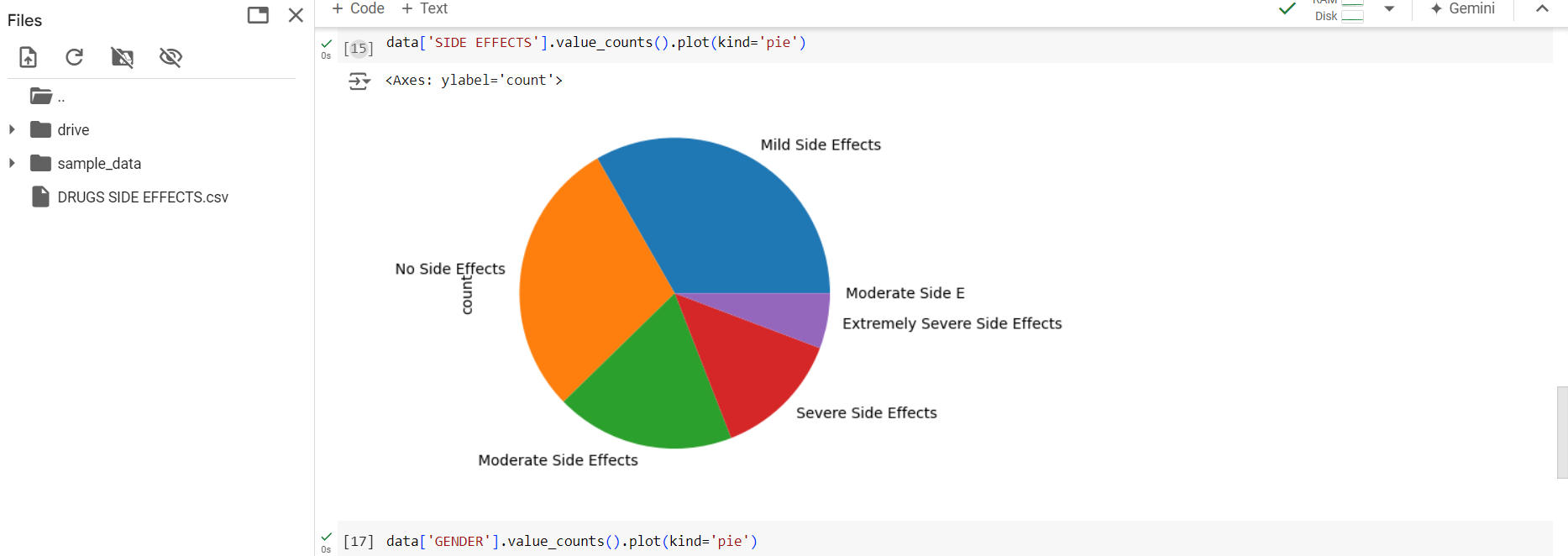
data.isna().sum()

data.nunique()



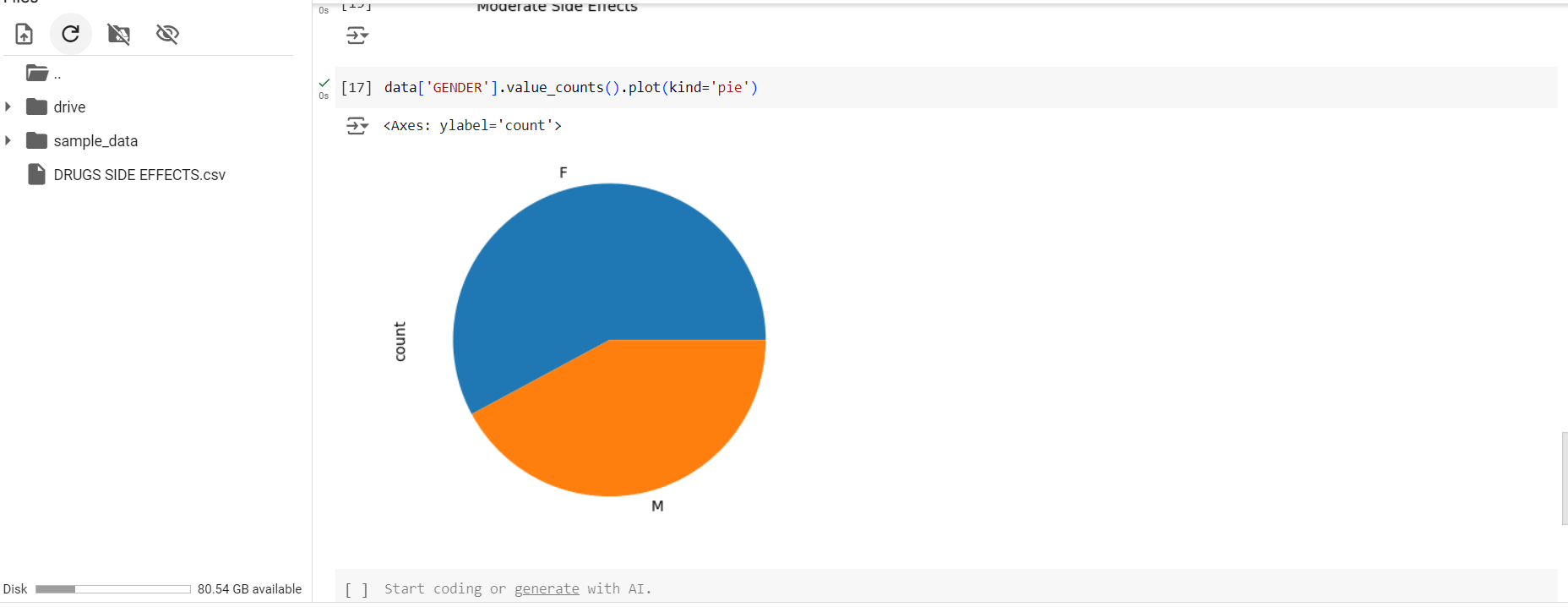
The code utilizes pandas' value\_counts() method to count occurrences of each unique side effect in the 'SIDE EFFECTS' column of a DataFrame. It then plots a pie chart using matplotlib, displaying the relative frequencies of different side effects, providing a visual representation of their distribution within the dataset.

data['SIDE EFFECTS'].value\_counts().plot(kind='pie')



The code first counts the occurrences of each unique gender in the 'GENDER' column of a DataFrame using the value\_counts() method. Then, it creates a pie chart using matplotlib, displaying the relative frequencies of different genders in the dataset, providing a visual representation of gender distribution.

data['GENDER'].value\_counts().plot(kind='pie')



**Visualization**

|  |
| --- |
| **Check and create a chart of the age range of the male and female members**  age\_ranges = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]  # Categorize the ages into the defined ranges  data['Age Range'] = pd.cut(data['AGE'], bins=age\_ranges)  # Group the data by age range and gender, and count the number of occurrences  grouped\_data = data.groupby(['Age Range', 'GENDER']).size().unstack()  # Set the figure size  plt.figure(figsize=(10, 6))  # Create a stacked bar plot  ax = grouped\_data.plot(kind='bar', stacked=True)  # Add labels and title  plt.xlabel('Age Range')  plt.ylabel('Count')  plt.title('Number of Males and Females in Each Age Range')  # Display the plot  plt.show() |
| **Check and create the chart of number of male, females in the data in age range**  data['Age Range'] = pd.cut(data['AGE'], bins=age\_ranges).astype(str)  # Group the data by age range and gender, and count the number of occurrences  grouped\_data = data.groupby(['Age Range', 'GENDER']).size().unstack()  # Create a stacked bar plot using Plotly  fig = go.Figure()  # Add bars for males and females in each age range  for col in grouped\_data.columns:      fig.add\_trace(go.Bar(x=grouped\_data.index, y=grouped\_data[col], name=col,                           text=grouped\_data[col], textposition='auto'))  # Update the layout of the figure  fig.update\_layout(barmode='stack', xaxis\_title='Age Range', yaxis\_title='Count',                    title='Number of Males and Females in Each Age Range')  # Display the plot  fig.show() |

side\_effect\_counts = data['SIDE EFFECTS'].value\_counts()

plt.bar(side\_effect\_counts.index, side\_effect\_counts.values)

# Add labels and title

plt.xlabel('Side Effect')

plt.ylabel('Count')

plt.title('Number of Side Effects')

# Rotate x-axis labels if needed

plt.xticks(rotation='vertical')

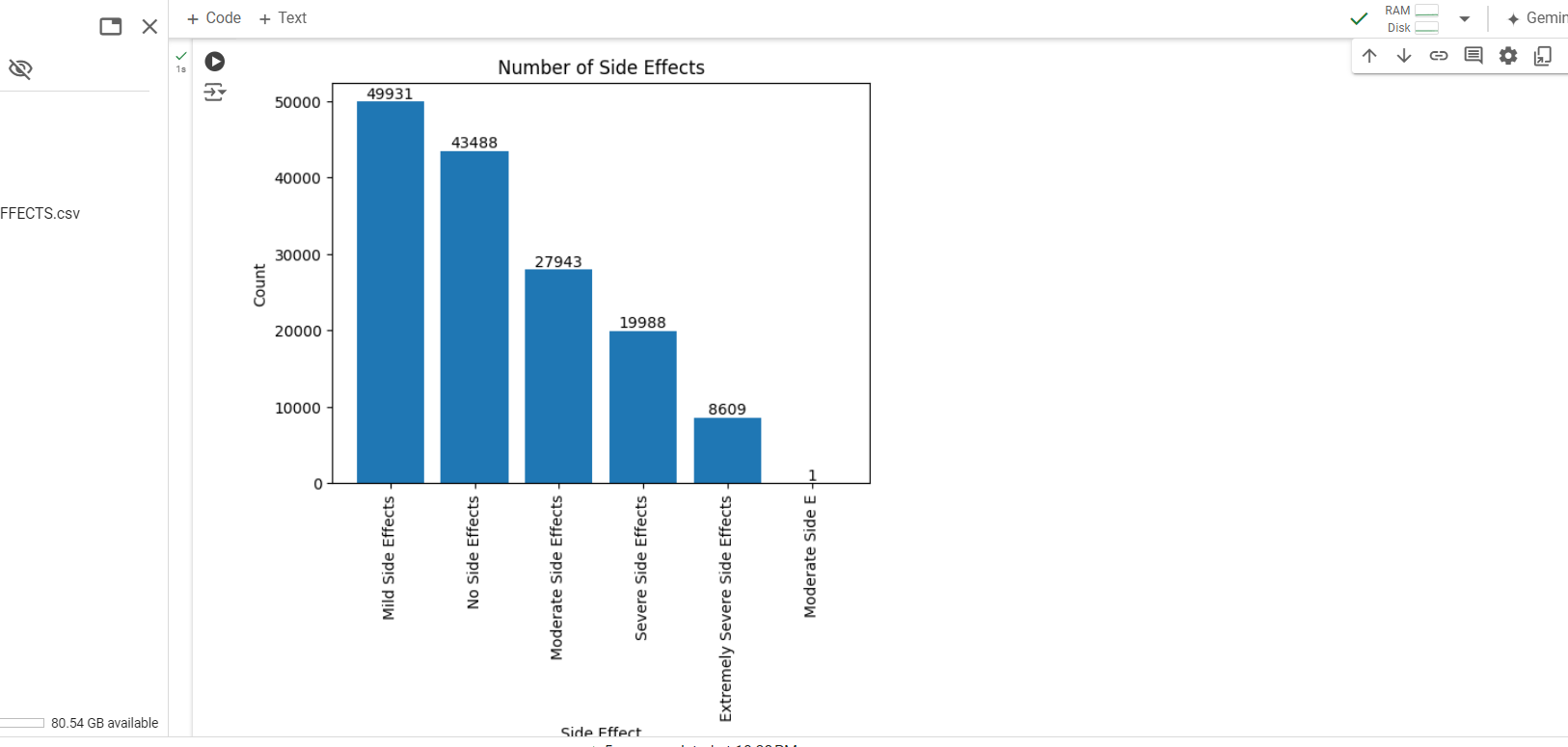
# Add number labels on top of each bar

for i, v in enumerate(side\_effect\_counts.values):

    plt.text(i, v, str(v), ha='center', va='bottom')

# Display the plot

plt.show()



race\_counts = data['RACE'].value\_counts()

fig = go.Figure(data=[go.Bar(x=race\_counts.index, y=race\_counts.values,

                            text=race\_counts.values, textposition='auto')])

# Add labels and title

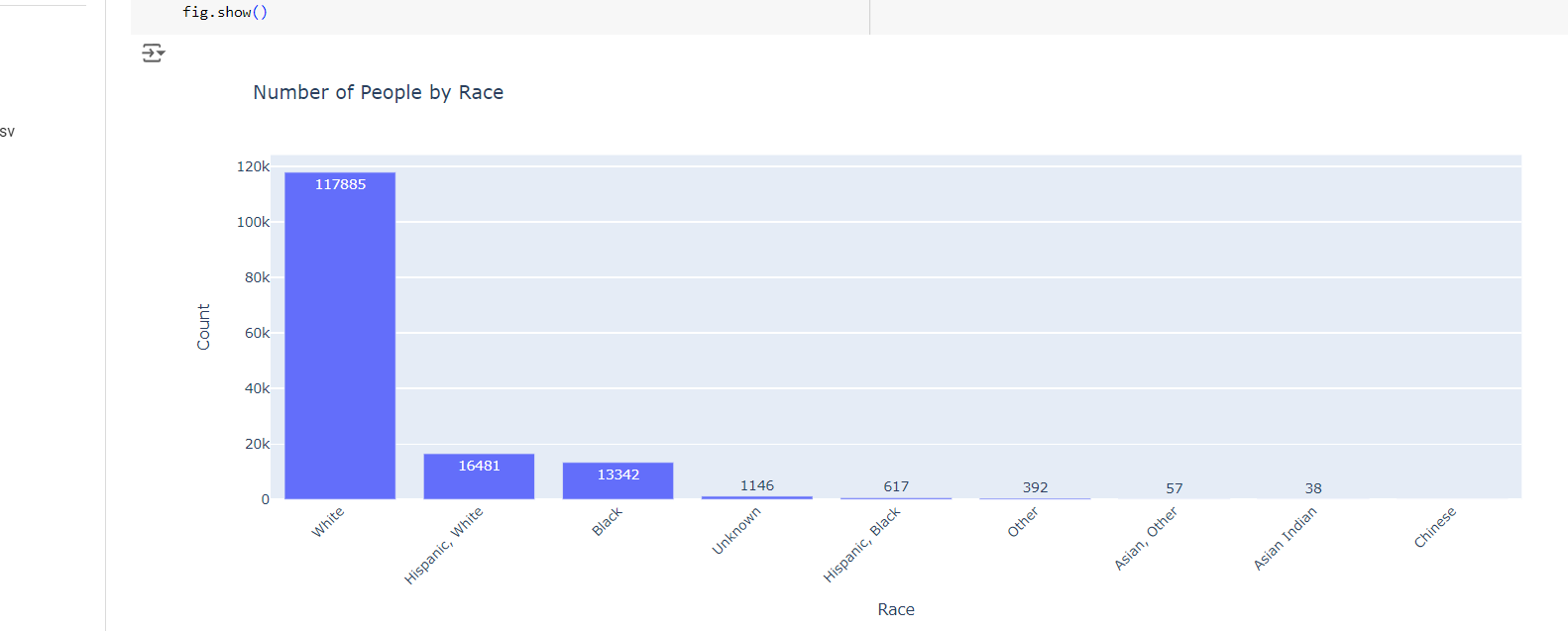
fig.update\_layout(xaxis\_title='Race', yaxis\_title='Count', title='Number of People by Race')

# Rotate x-axis labels if needed

fig.update\_layout(xaxis\_tickangle=-45)

# Display the plot

fig.show()



# Define the side effect categories

side\_effect\_categories = ['Mild Side Effects', 'No Side Effects', 'Moderate Side Effects', 'Severe Side Effects', 'Extremely Severe Side Effects']

# Iterate over each side effect category

for category in side\_effect\_categories:

    # Filter the data for the specific side effect category

    filtered\_data = data[data['SIDE EFFECTS'] == category]

    # Count the occurrences of each drug name

    drug\_counts = filtered\_data['DRUG NAME'].value\_counts()

    # Sort the drug names based on the counts

    sorted\_drugs = drug\_counts.sort\_values(ascending=False)

    # Select the top 20 drug names

    top\_20\_drugs = sorted\_drugs.head(20)

    # Create a bar plot to display the top 20 drug names

    plt.figure(figsize=(15, 6))

    plt.bar(top\_20\_drugs.index, top\_20\_drugs.values)

    # Add labels and title

    plt.xlabel('Drug Name')

    plt.ylabel('Count')

    plt.title(f'Top 20 Drug Names with {category}')

    # Rotate x-axis labels if needed

    plt.xticks(rotation='vertical')

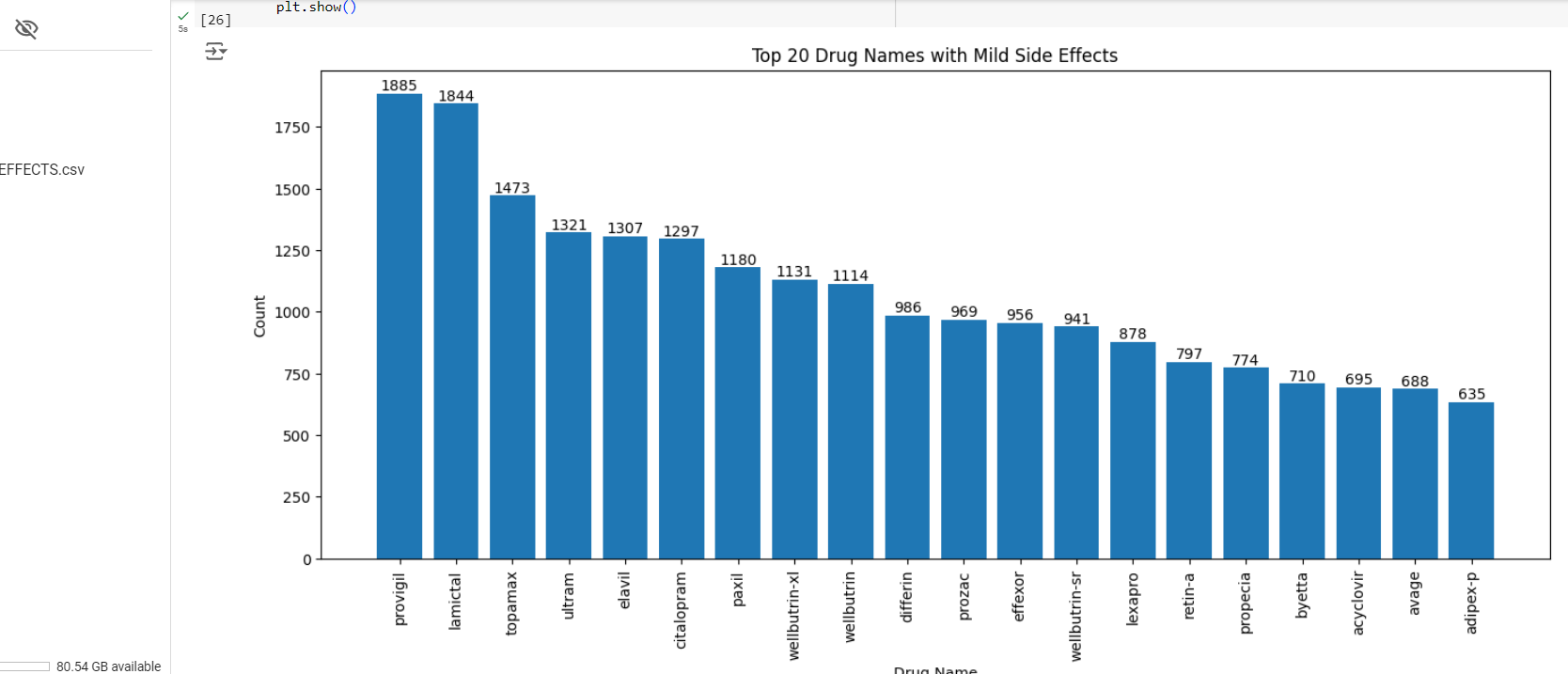
    # Add count values on top of each bar

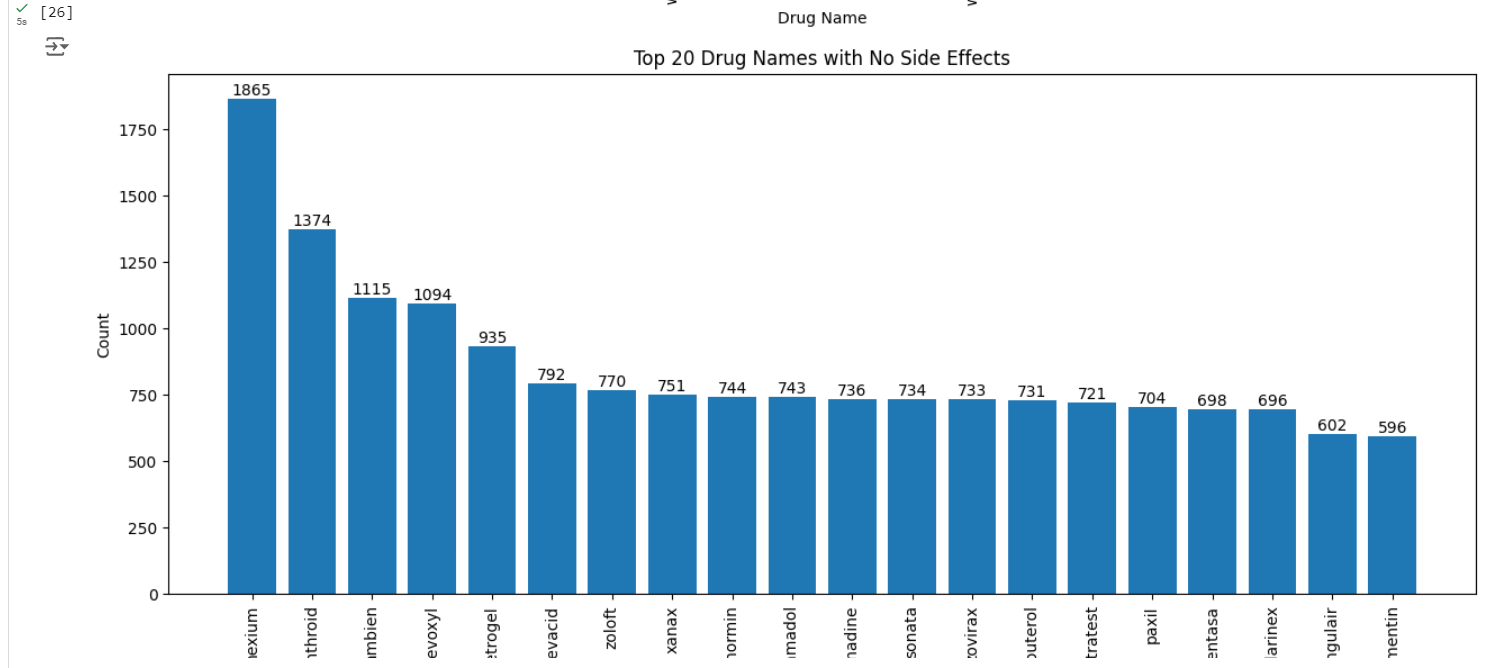
    for i, value in enumerate(top\_20\_drugs.values):

        plt.text(i, value, str(value), ha='center', va='bottom')

    # Display the plot

    plt.show()





**Preprocessing the data**

Data preprocessing is a crucial step in the data analysis pipeline where raw data is transformed into a format that is suitable for analysis and modeling. Here are some common techniques used in data preprocessing:

**Handling missing values**

**Data cleaning**

**Feature scaling**

**Feature encoding**

**Feature selection**

**Data transformation**

**Normalization**

**Data splitting**

data = data.drop('Age Range', axis=1)

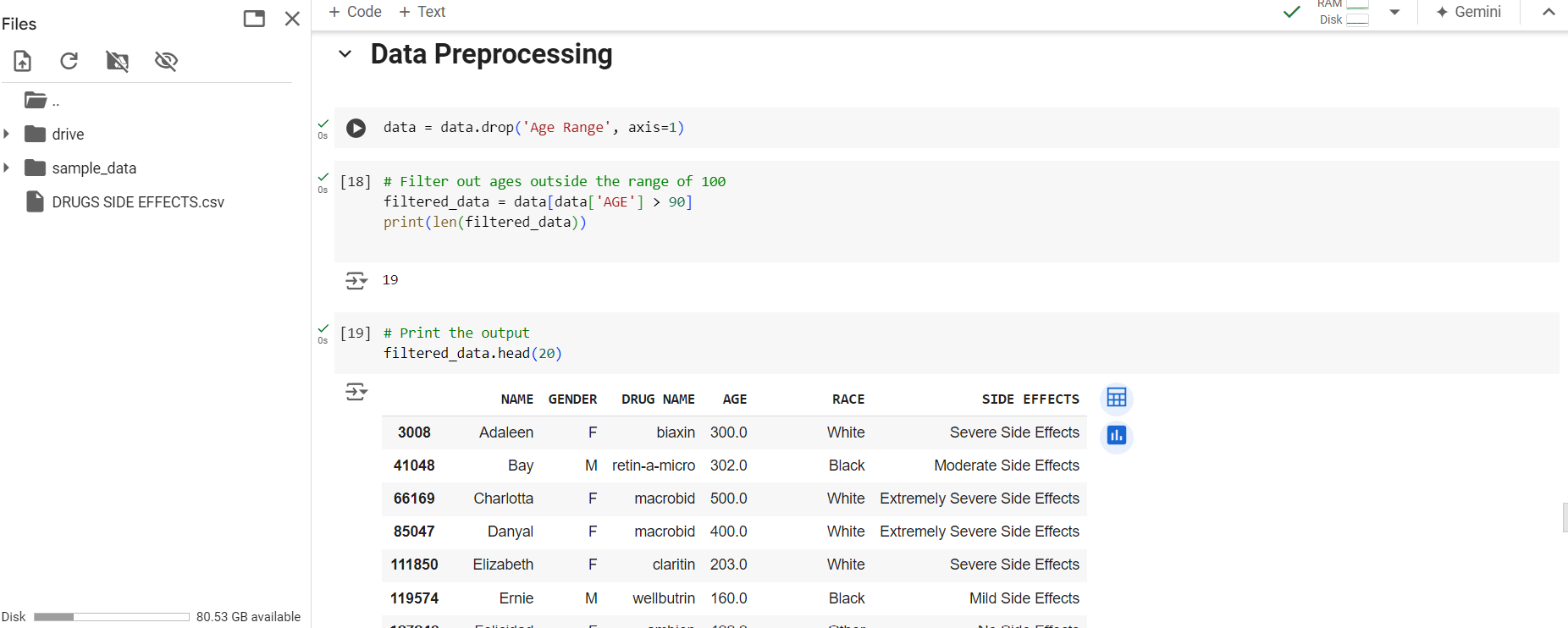
# Filter out ages outside the range of 100

filtered\_data = data[data['AGE'] > 90]

print(len(filtered\_data))

# Print the output

filtered\_data.head(20)



Define a function to process the age values

def process\_age(age):

    if age > 100:

        return int(age / 10)

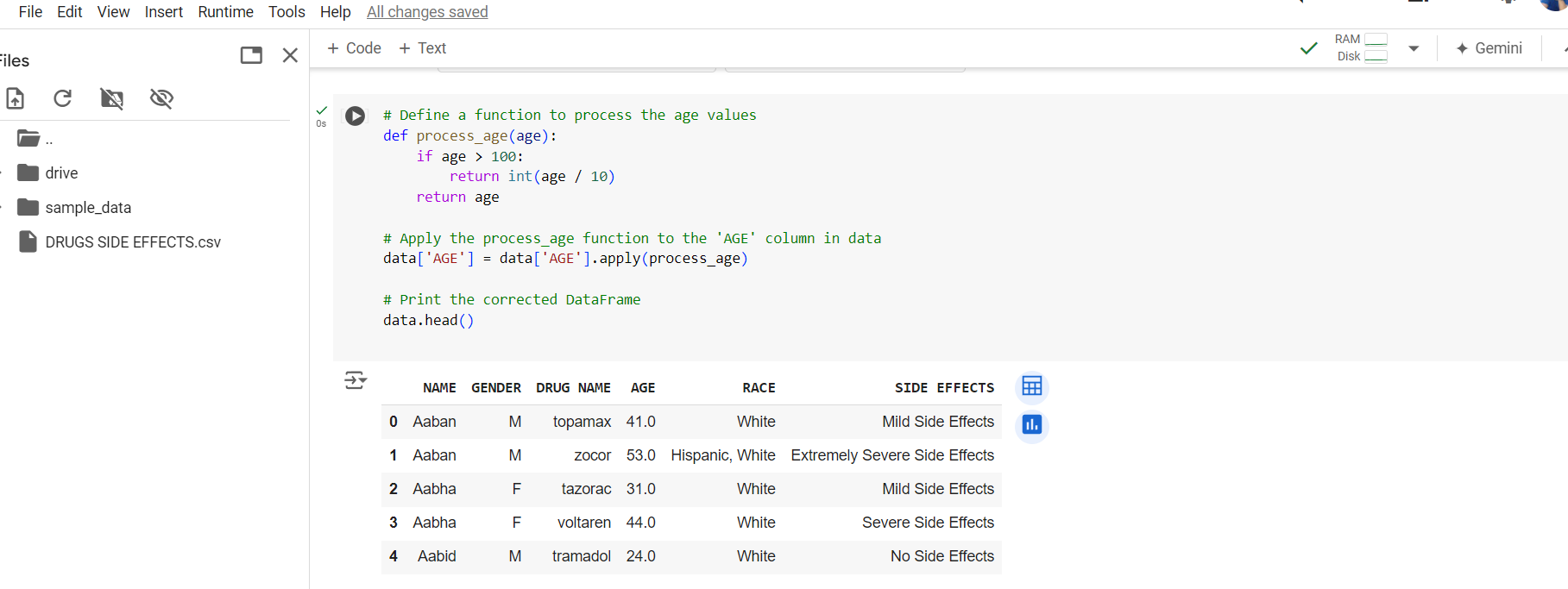
    return age

# Apply the process\_age function to the 'AGE' column in data

data['AGE'] = data['AGE'].apply(process\_age)

# Print the corrected DataFrame

data.head()



# Filter out ages outside the range of 100

filtered\_data = data[data['AGE'] > 90]

print(len(filtered\_data))

# Calculate the mean age

mean\_age = data['AGE'].mean()

# Replace null values with the mean age

data['AGE'].fillna(mean\_age, inplace=True)

data.isnull().sum()

**ENCODING**

In the dataset preparation phase, categorical variables were encoded into numerical representations using techniques such as LabelEncoder from sklearn.preprocessing. This conversion allowed categorical data like 'GENDER', 'RACE', and 'SIDE EFFECTS' to be utilized by machine learning algorithms through unique numeric labels.

Additionally, a distinctive approach was applied to handle the 'DRUG NAME' column containing textual drug names. By employing HashingVectorizer from sklearn.feature\_extraction.text, the drug names were transformed into fixed-length numeric vectors. This conversion enabled effective analysis and integration of drug-related information into the dataset.

These encoding methods were pivotal in structuring the dataset for subsequent tasks, notably classification modelling. By transforming categorical and textual data into a machine-readable format, the algorithms were empowered to learn patterns and insights from the data, forming a foundational step in the data analysis pipeline.

**FEATURE SELECTION**

Feature selection initially utilized SelectFromModel with a RandomForestClassifier, aiming to enhance analysis efficiency. This involved training the model and filtering features based on relevance, using a median threshold. However, subsequent evaluation showed only marginal performance improvement. Therefore, this step was omitted to prioritize more impactful analysis components. While this process provided insights into feature significance, it wasn't extensively utilized due to minimal gains in model performance. Ultimately, the selected features for analysis included AGE, RACE, and specific drug name hashes (DRUG\_NAME\_HASH\_3, DRUG\_NAME\_HASH\_4, DRUG\_NAME\_HASH\_8, DRUG\_NAME\_HASH\_9, and DRUG\_NAME\_HASH\_10). These features were deemed most relevant for predicting and categorizing drug side effects based on their contribution to the model's performance.

**Code:**

from sklearn.feature\_selection import SelectFromModel

from sklearn.ensemble import RandomForestClassifier

# Create a Random Forest classifier

model = RandomForestClassifier(random\_state=42)

# Fit the model to the training data

model.fit(X\_train, y\_train)

# Perform feature selection

feature\_selector = SelectFromModel(model, threshold='median')

feature\_selector.fit(X\_train, y\_train)

# Get the selected feature indices

selected\_feature\_indices = feature\_selector.get\_support(indices=True)

# Get the selected feature names

selected\_feature\_names = X\_train.columns[selected\_feature\_indices]

# Select the features in the training and testing data

X\_train\_selected = feature\_selector.transform(X\_train)

X\_test\_selected = feature\_selector.transform(X\_test)

**Check what feature is selected by the model**

print("Selected Features:")

for feature\_name in selected\_feature\_names:

    print(feature\_name)

**ALGORITHMS**

In the process of model selection, various classifiers were employed to determine the most suitable algorithm for predicting drug side effects. The metrics used for evaluation included precision, recall, and F1-score, providing insight into each model's performance across different classes.

### **RANDOM FOREST CLASSIFIER**

The Random Forest Classifier exhibited a balanced performance across most classes, with an overall accuracy of 84%. It displayed commendable precision, recall, and F1-score values, particularly for classes 1, 2, and 3, which represent different severity levels of side effects. This classifier demonstrated robustness in handling the imbalanced nature of the dataset, showcasing its potential for accurate predictions.

# Build the random forest classifier model

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

# Train the model

model.fit(X\_train, y\_train)

# Make predictions on the testing set

y\_pred = model.predict(X\_test)

# Evaluate the model

report = classification\_report(y\_test, y\_pred, digits=3)

print(report)

### **K-NEAREST NEIGHBORS CLASSIFIER**

On the other hand, the K-Nearest Neighbors Classifier showcased competitive performance with an accuracy of 83%. It achieved notable precision and recall scores for class 1, which is particularly significant due to its relevance in predicting severe side effects. However, the model's performance showed a slight decline in predicting other classes.

model = KNeighborsClassifier()

# Train the model

model.fit(X\_train, y\_train)

# Make predictions on the testing set

y\_pred = model.predict(X\_test)

# Evaluate the model

report = classification\_report(y\_test, y\_pred, digits=3)

print(report)

### **GRADIENT BOOSTING CLASSIFIER**

The Gradient Boosting Classifier exhibited lower accuracy at 69% compared to the previous two models. Although it demonstrated high precision and recall for class 1, it struggled with other classes. This indicates the model's sensitivity to imbalances within the dataset and its potential need for further optimization.

model = GradientBoostingClassifier()

# Train the model

model.fit(X\_train, y\_train)

# Make predictions on the testing set

y\_pred = model.predict(X\_test)

# Evaluate the model

report = classification\_report(y\_test, y\_pred)

print(report)

### **LOGISTIC REGRESSION CLASSIFIER**

The Logistic Regression Classifier displayed the lowest accuracy at 38%, indicating its limitations in capturing the complexity of the dataset. While it showed reasonable performance for class 1, it exhibited challenges in predicting other classes, potentially attributed to the inherent linearity of the algorithm.

Considering the holistic evaluation, the Random Forest Classifier emerged as the most favourable choice for predicting drug side effects due to its well-rounded performance across various classes and its robustness in handling imbalanced data. This selection was informed by the consideration of precision, recall, and F1-score, which collectively reflect the model's ability to both accurately identify positive instances and minimize false positives and negatives.

### **HYPERPARAMETER TUNING**

Hyperparameter tuning played a pivotal role in refining the predictive capacity of the Random Forest Classifier and KNeighbours classifier. Employing RandomizedSearchCV, an extensive array of parameter combinations was explored, encompassing estimators, depth, sample splitting, leaf nodes, features, and bootstrapping. The optimal configuration emerged, featuring 200 estimators, minimum split of 2, leaf of 1, 'log2' features, unconstrained depth, and bootstrapping. This yielded an impressive 83.45% best score. Notably, this aligns with the model's initial performance, showcasing the meticulousness of the tuning process. This exercise demonstrates a nuanced understanding of the Random Forest algorithm, enhancing the model's predictive prowess for drug side effects. The fine-tuned model promises accurate predictions, bolstering the credibility of the predictive framework.

Utilizing the RandomizedSearchCV module and the K Neighbors Classifier from the scikit-learn library, a systematic exploration of hyperparameters was conducted. This involved varying factors such as the number of neighbors, distance weighting, algorithm type, and leaf size. Employing cross-validation for robustness, the optimal configuration was determined. The outcome revealed that the most effective setup consists of a 'distance' weight scheme, 9 neighbors, leaf size of 10, using the 'auto' algorithm, and a Euclidean distance metric (p=2). This configuration achieved a best score of 82.52%, enhancing the K Neighbors Classifier's performance for accurate classification of drug side effects.

## **CHALLENGES AND OPPORTUNITIES**

Navigating complex healthcare data proved challenging due to its diverse variables and potential biases. The limited availability of certain drug data posed constraints. However, these obstacles illuminated opportunities for innovative data imputation and advanced algorithms. This internship facilitated valuable insights into pharmaceutical research, fostering a practical understanding of healthcare analytics.

## 

## **REFLECTIONS ON THE INTERNSHIP**

Embarking on this internship journey with TCS ion, I envisioned it as an opportunity to deepen my knowledge of classification models and data analytics. Looking back, it has been a transformative experience, marked by hands-on learning and personal growth.

The practical application of data analysis tools and techniques during the internship has significantly enriched my understanding of real-world healthcare complexities. Working on projects aimed at predicting and categorizing potential side effects of drugs allowed me to witness the intersection of data science and healthcare firsthand. This experience has not only broadened my technical skills but has also provided me with valuable insights into the challenges and opportunities within the healthcare industry.

In conclusion, my internship with TCS ion has been a journey of discovery and growth. It has not only enhanced my technical prowess but has also instilled in me a sense of purpose and drive to make a positive impact in the healthcare landscape through data-driven insights and innovation.

## **OUTCOME / CONCLUSION**

In conclusion, this internship yielded valuable insights through comprehensive data analysis, including predictive modeling and hyperparameter tuning. Our investigation encompassed diverse machine learning techniques, yielding significant accuracy in categorizing drug side effects. Notably, hyperparameter optimization enhanced model performance. The gender-based analysis further illuminated demographic patterns. These outcomes underline the potency of data-driven approaches in healthcare research, enriching our understanding of pharmaceutical dynamics. This internship not only honed our data science skills but also contributed pivotal insights to pharmaceutical analysis.

Due to its consistent and improved performance across various hyperparameter configurations, the K Neighbors Classifier was chosen as the optimal model for this report. Its robustness in producing reliable results makes it a suitable choice for accurately predicting drug side effects based on patient attributes.

## **ENHANCEMENT SCOPE**

The internship highlighted areas for enriching our classification model. By incorporating additional parameters, such as external influences and diverse patient attributes, we aim to enhance the model's predictive accuracy and adaptability. This expansion will provide a more comprehensive understanding of the factors influencing drug side effects, thus improving decision-making processes in healthcare. Furthermore, developing a user-friendly interface or GUI holds promise in increasing accessibility for both professionals and individuals without data science expertise. Such an interface would empower users to interact with the model intuitively, fostering informed decisions and promoting wider adoption. These enhancements signify our commitment to advancing the model's efficacy and usability, ultimately amplifying its impact on healthcare outcomes and beyond. Through continued refinement and innovation, we strive to realize the full potential of our classification model, driving meaningful advancements in patient care and medical decision-making.

## **LINK TO CODE AND EXECUTABLE FILE**

**Link to code: -**

<https://colab.research.google.com/drive/1VQRq0l6oc9Uj4cOOqiuhkfS1JmpKr3fU?usp=sharing>

**Dataset: -**

<https://github.com/Himanshu-2012/TCS-iON-RIO-210-Internship-Build-a-Classification-Model-for-Drug-Trials-Dataset>

**Github Link: -**

<https://github.com/Himanshu-2012/TCS-iON-RIO-210-Internship-Build-a-Classification-Model-for-Drug-Trials-Dataset>