|  |  |
| --- | --- |
| Name Of The Student | Himanshu |
| Internship Project Topic | TCS iON RIO-210: Build a Classification Model for Drug Trials Dataset |
| Name of the Organization | TCS iON |
| Name of the Industry Mentor | Himdweep Walia |
| Name of the Institute | Amity University |

|  |  |  |
| --- | --- | --- |
| Date | Day # | Hours Spent |
| 31-05-2024 | Day-38 | 6 Hours |
| Activities done during the day:  **Project Hands-on – Explore the data By following Exploratory Data Analysis**  **Link of the google drive google Colab file :-**  <https://colab.research.google.com/drive/1VQRq0l6oc9Uj4cOOqiuhkfS1JmpKr3fU?usp=sharing>  **EDA**  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') | | |
|  | | |