

Name: **Kanaishk Garg**

ID: [rishi.gupta90@gmail.com](mailto:rishi.gupta90@gmail.com)

Project: **Customer Service Requests Analysis Project-1**

### **Summary:**

1. Load and Check the Data

In this step, we read in the data from a csv file using pandas, and checked the shape and basic information about the dataset.

2. Data Cleaning

In this step, we performed several cleaning tasks such as dropping irrelevant columns, dropping missing values, and converting data types where appropriate.

3. Exploratory Data Analysis

In this step, we used seaborn and matplotlib to visualize the data, and answer questions such as what are the top complaints, how do complaint types vary across cities, and how does the time to resolve complaints vary across cities and complaint types.

4. Visualizing Complaints by City

In this step, we used a bar chart to show the number of complaints by city and complaint type, and used color to distinguish between different complaint types.

5. Analyzing Response Time

In this step, we converted the time elapsed column to hours, and visualized the average response time using a KDE plot.

6. Statistical Analysis

In this step, we used statsmodels to perform a linear regression and identify significant variables based on p-values.

7. Hypothesis Testing

In this step, we used scipy to perform a Kruskal-Wallis H test to determine whether the average response time varies significantly across different complaint types. We failed to reject the null hypothesis that all sample distributions are equal.

### **Working Steps:**

1. Explore the dataset

Read the CSV file using pandas read\_csv function

Explore the dataset using pandas functions like head(), info(), describe() and isnull().sum()

2. Data Cleaning

Drop unnecessary columns from the dataset

Replace missing values with appropriate values

Convert columns to appropriate data types

3. Data Analysis and Visualization

3.1 Visualize the distribution of complaints by the hour of the day

- Convert created\_date column to datetime format
- Create a new column with the hour of the day
- Plot the distribution of complaints by hour using matplotlib

- 3.2 Visualize the distribution of complaints by day of the week
  - Create a new column with the day of the week
  - Plot the distribution of complaints by day of the week using matplotlib
- 3.3 Visualize the top 10 complaint types
  - Use value\_counts() to get the top 10 complaint types
  - Plot a bar chart using matplotlib
  - Add labels and save the plot
  - Convert to seaborn plot
- 3.4 Display the various types of complaints in each city
  - Create a pivot table with cities as columns and complaint types in rows
  - Plot a heatmap using seaborn
- 3.5 Create a DataFrame, df\_new, which contains cities as columns and complaint types in rows
  - Create a pivot table with cities as columns and complaint types in rows
4. Visualize the major types of complaints in each city
  - 4.1 Draw another chart that shows the types of complaints in each city in a single chart, where different colors show the different types of complaints
    - Create a stacked bar chart using pandas
  - 4.2 Sort the complaint types based on the average Request\_Closing\_Time grouping them for different locations
    - Group data by complaint type and city and calculate the average time taken to close a request
    - Sort the data based on the average time taken to close a request
    - Plot the data as a grouped bar chart using seaborn
    - Add labels and save the plot
    - Customize the legend
5. See whether the average response time across different complaint types is similar (overall)
  - 5.1 Visualize the average of Request\_Closing\_Time
    - Convert time elapsed column to hours from seconds
    - Plot a kernel density estimate (KDE) plot using seaborn
6. Identify the significant variables by performing statistical analysis using p-values
  - Use statsmodels to perform a multiple linear regression and calculate the p-values
7. Perform a Kruskal-Wallis H test
  - 7.1 Fail to reject H0: All sample distributions are equal
    - Use scipy to perform a Kruskal-Wallis H test