Dec’22 – Jun’23 Internship Report

Automation of Data Pre-processing Techniques using ML

Guided by:

Shri T. Venkatsubramanian (Industrial mentor)
Mr. Harikrishna Rangam (Project mentor)
RBG Labs, Department of Engineering Design
IIT Madras, Chennai 600036

Submitted by:

SURAJ || ED19B062
Department of Engineering Design
IIT Madras, Chennai 600036
E-mail: Ed19b062@smail.iitm.ac.in

CERTIFICATE

This is to certify that the project titled **Automation of Data Pre-Processing techniques using Machine Learning** submitted by **Suraj Ahirwar**, to the Indian Institute of Technology Madras, Chennai for the award of the degree of Bachelor of Technology in Engineering Design and Master of Technology in Automotive/Biomedical Engineering, is a bona fide record of the research work done by him under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

…………………………….

Project Advisor:
Sri. T Venkat Subramaian
Advisor Centre for excellence in Road Safety Chennai 600036

Signature:

Place: IIT Madras, Chennai

Date:

ACKNOWLEDGEMENTS

I am sincerely grateful for the opportunity to intern at CoERS company and would like to express my heartfelt appreciation to my mentor, T Venkat Subramanian. I am truly thankful for the opportunity to work with such a dynamic and accomplished team at CoERS company. The hands-on experience and exposure to real-world projects have been instrumental in shaping my professional growth. I am also thankful to Department of Engineering Design at IIT Madras, for providing me with a solid foundation. Thank you all for your invaluable support and guidance during my internship.

ABSTRACT

The growing volume and complexity of data pose significant challenges in the data pre-processing phase of machine learning projects. Manual data cleaning and transformation processes are time-consuming, error-prone, and hinder the overall efficiency of the data analysis pipeline. To address these challenges, we present an automated data pre-processing solution that leverages the power of machine learning techniques.

In this report we present an automated data pre-processing solution that utilizes machine learning (ML) techniques to address the challenges of data cleaning and transformation. Our approach incorporates ML algorithms for handling missing values, outliers, and inconsistent data. It also includes feature selection, dimensionality reduction, and data normalization for accurate modeling.

The proposed solution reduces manual effort, minimizes errors, and improves model accuracy. It streamlines the data analysis workflow, allowing data scientists to focus on modeling and analysis. We discuss the methodology, implementation details, and evaluation results, showcasing the effectiveness and practical applications of our solution.

By automating data pre-processing, our project contributes to the advancement of ML techniques. It accelerates the development and deployment of high-quality models, enabling efficient data analysis in real-world applications.

Table of Contents Page No

Problem Statement

The Centre of Excellence for Road Safety aims to improve road safety and reduce accident rates through data-driven decision-making. However, the current manual data processing methods present limitations in terms of efficiency and accuracy. Manual processing is time-consuming, error-prone, and leads to incomplete or inconsistent datasets. This hinders the organization's ability to derive accurate insights, identify accident-prone areas, and develop targeted interventions. There is a need for an automated data pre-processing solution that can effectively handle challenges such as missing data, outliers, and inconsistencies. By addressing these challenges, the Centre of Excellence for Road Safety can enhance its analytical capabilities, provide evidence-based recommendations, and contribute to the reduction of road accidents.

OBJECTIVES

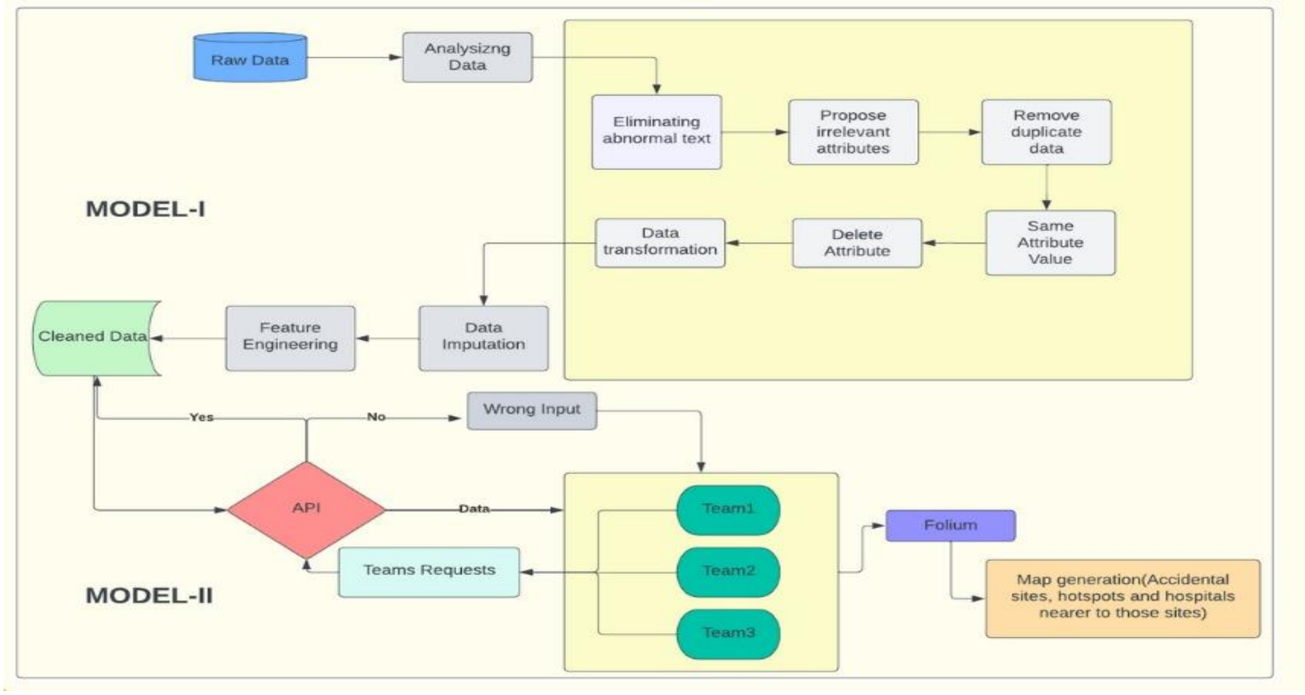
The objective of the project "Automation of Data Preprocessing Techniques using Machine Learning" at the Centre of Excellence for Road Safety is to develop an advanced solution that automates data preprocessing, enhances data quality, increases productivity, enables data-driven insights, facilitates collaboration, and ensures scalability. By leveraging machine learning algorithms, the project aims to streamline and automate tasks such as handling missing data, outlier detection, normalization, and feature selection. This will reduce manual effort, improve the accuracy and reliability of the preprocessed data, and save time. The system will empower researchers and analysts to make informed decisions based on trustworthy data, uncover hidden patterns, and promote collaboration among stakeholders. Furthermore, the solution will be scalable and adaptable, capable of handling diverse datasets and accommodating future growth. The objective is to support the Centre of Excellence in its mission to enhance road safety and decrease accident rates through data-driven decision-making.

PART 1: MODEL DEVELOPMENT

1.1 Introduction:

The project "Automation of Data Pre-processing Using ML" focuses on addressing challenges associated with road accident data. The objective is to develop an automated solution that effectively preprocesses the data, ensuring its accuracy and usability. By leveraging machine learning techniques, the project aims to correct errors, handle abnormal text, resolve location-related issues, and impute missing coordinates. The refined data becomes a valuable resource for generating hotspots, visualizing accident sites, and conducting further analysis. Additionally, the project includes the development of APIs and a user-friendly web framework, allowing users to upload and process their own data. The processed data can be conveniently downloaded, empowering users to utilize the refined data for decision-making in areas like traffic management and urban planning. This project offers a comprehensive solution for enhancing the quality and usability of road accident data, facilitating data-driven decision-making in various domains.

Model Designing Approach:



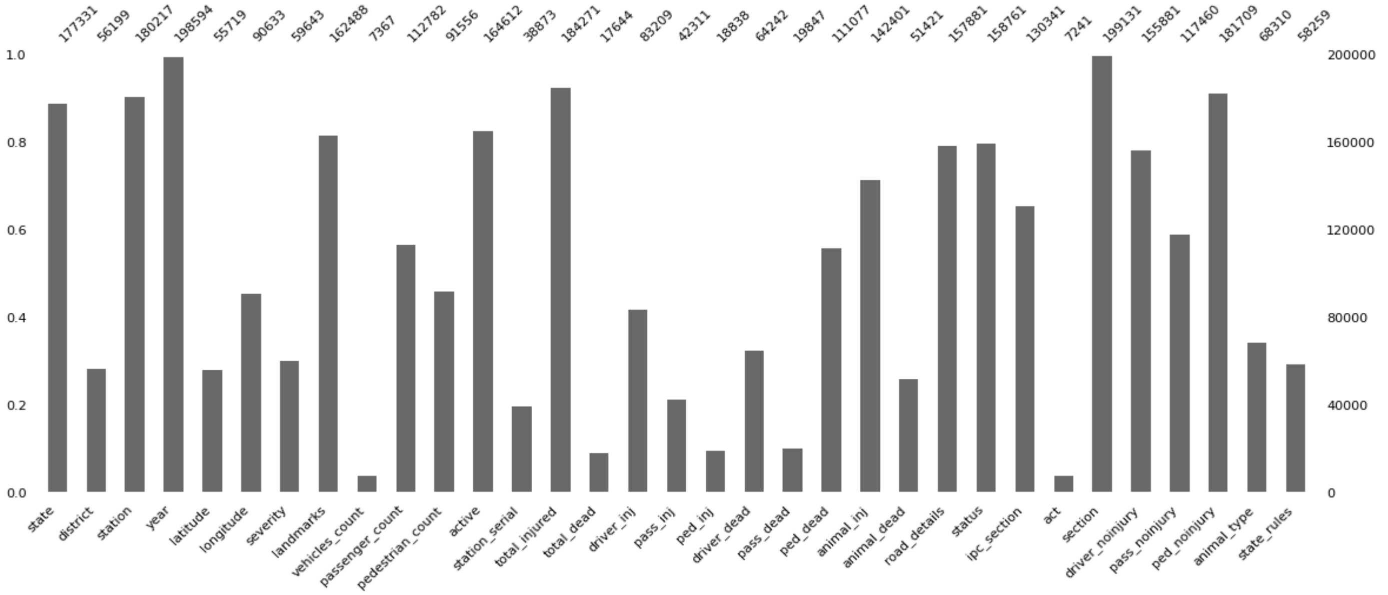
1.2 Data Cleaning:

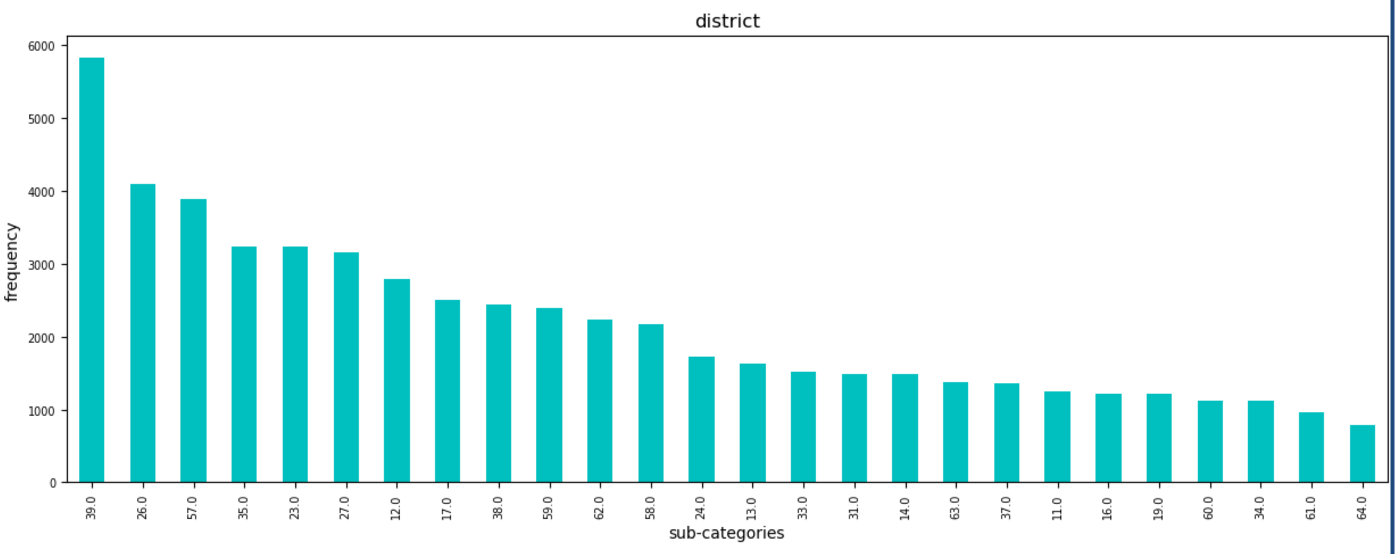
The data cleaning process in this project follows a systematic approach to ensure the accuracy and quality of the input dataset and is mostly oriented to the road accidental datasets.

- **Data Analysis:** In the data analysis step, the road accident dataset is thoroughly examined to identify its structure and content. This analysis helps in understanding the types of data variables present, such as numerical, categorical, or textual, which aids in subsequent cleaning steps. It also involves identifying any inconsistencies or missing values in the dataset and laying the foundation for further cleaning and validation procedures.
- **Abnormal Text and Irrelevant Features:** Abnormal text entries and irrelevant features are systematically removed from the dataset, ensuring that they do not impact the subsequent analysis. Cells or values considered irrelevant are set to None, effectively eliminating their influence on the data and preventing any misleading interpretations.
- **Percentage-based Validation:** To ensure data quality, a percentage-based validation approach is applied to specific columns. For example, in the pincode column, a defined criterion of being an integer or float with a length of six digits is checked. Values that do not meet this criterion are categorized as abnormal and set to None. This approach allows for the identification and handling of values that deviate significantly from the expected range or pattern, improving the overall integrity of the dataset.
- **Handling Sensitive Columns, Address Reshaping, and State/District Validation:** Sensitive columns such as Latitude, Longitude, Pincode, State, and District, station undergo a thorough cleaning process to ensure consistency and accuracy in the data. Cleaning and formatting techniques are applied to these columns, enabling standardization and enhancing the precision of subsequent analyses. The address column is subjected to a specific formatting procedure, reshaping the data and ensuring uniformity across entries. State and district values are validated against predefined Indian state and district names to ensure their correctness and relevance. Values that do not match the predefined criteria are set to None, maintaining data consistency and eliminating erroneous entries.
- **Removing outliers:** After handling the None values, the project focuses on removing outliers in the non-sensitive columns to ensure the integrity and accuracy of the data. Outliers can significantly affect data analysis and modeling results, so it is crucial to identify and appropriately deal with them. The removal of outliers follows a statistical approach, with a specific emphasis on numerical columns. Machine learning algorithms, such as the Random Forest Regressor, are utilized for outlier detection and removal. The Random Forest Regressor algorithm employs decision trees to analyze the relationships between variables and predict the target variable. During this process, outliers are identified as data points that deviate significantly from the expected patterns. By leveraging the capabilities of the Random

Forest Regressor algorithm, outliers are detected and subsequently eliminated from the dataset. This step ensures a more accurate representation of the non-sensitive columns, improving the overall data quality and reliability. Through the removal of outliers, the project strives to enhance the robustness of the dataset, enabling more accurate and reliable analysis and modeling outcomes. By eliminating extreme values that could distort the results, the project ensures that the subsequent data processing and analysis tasks are built upon a solid foundation.

1.3 Graphs showing null values and sub-categories present:





1.4 Dealing with None values present in the data:

After the initial data cleaning steps, the project focuses on handling None values in the dataset. The user is prompted to specify the final columns they want to transform. The data is divided into two categories: sensitive columns (such as state, latitude, longitude, Pincode, and district) and non-sensitive columns. Sensitive columns require a different approach for imputation as they cannot be filled based on average or frequency methods.

For the imputation of sensitive columns, a multi-step process is employed. Initially, dependent columns are filled using information from other related columns. For instance, if a pincode column has None values, the corresponding address column cells are analyzed to extract the pin code information and fill in the missing values. Similarly, if state or district information is present in the address column, the corresponding columns are checked and filled accordingly.

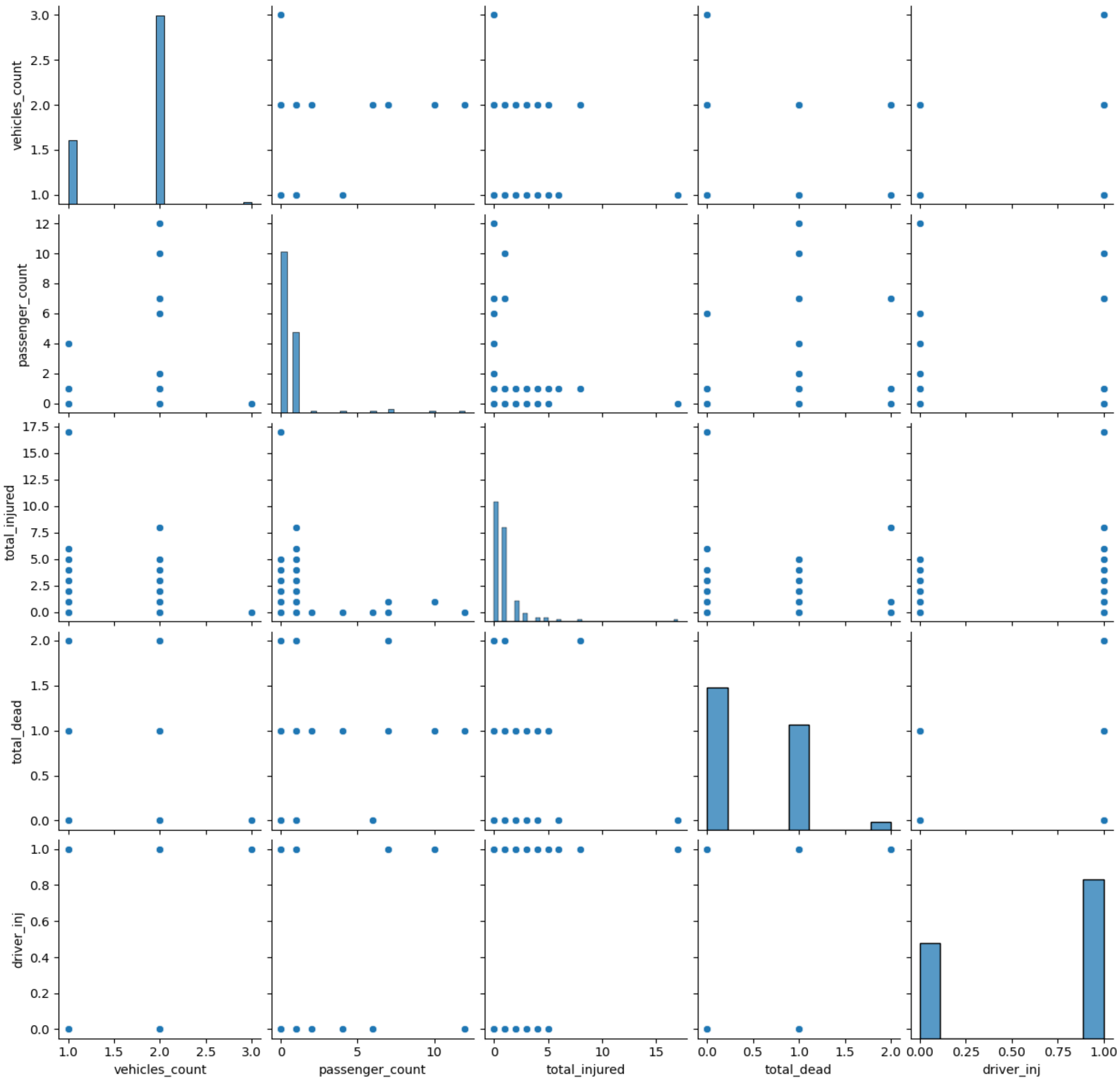
To handle the imputation of sensitive columns, advanced methodologies such as geocoding and reverse geocoding are applied. Geocoding provides precise coordinates for a location, allowing for the filling of None values in latitude and longitude columns. Reverse geocoding is employed to fill None values in state, district, pin code, and other related columns. Additionally, the connection between districts and states is verified to ensure accurate filling based on reverse geocoding results.

Moving on to the imputation of non-sensitive data columns, two approaches are used: Random Forest Regressor and Simple Imputer. The Random Forest Regressor employs a machine-learning technique that utilizes decision trees to predict missing values based on the values present in other columns. On the other hand, the Simple Imputer fills in missing values based on statistics such as the median or most frequent value of the respective column.

After obtaining the imputed values from both the Random Forest Regressor and Simple Imputer, the model compares them and fills the values close to the median or most frequent value for each specified column.

By implementing these methodologies, the project effectively handles None values in the dataset, ensuring that the final data is more complete and suitable for further analysis and visualization tasks.

1.5 Covariance matrix – understanding the features dependency



The covariance matrix graph illustrates the relationships between variables in our dataset, such as vehicles count, passengers count, total injured, total dead, and driver injured. It provides insights into the direction and strength of these relationships. Positive covariance values indicate a positive relationship, while negative values suggest a negative relationship. By examining this graph, we can identify patterns and dependencies among the variables, helping us make informed decisions and gain valuable insights for accident analysis and risk assessment.

1.6 Data Transformation: Preparing the Dataset for Analysis and Visualization

In the data transformation stage, the project focuses on preparing the dataset for analysis, modeling, and visualization. This involves applying various techniques such as feature engineering, normalization, dimensionality reduction, aggregation, data encoding, and standardization.

Feature engineering techniques are used to create new features or modify existing ones to capture meaningful information.

Through comprehensive data transformation, the project optimizes the dataset for analysis, modeling, and visualization, facilitating the extraction of valuable insights and informed decision-making.

1.7 ML Concepts Used in the Model: Enhancing Data Analysis and Predictive Modeling

The project incorporates various machine learning (ML) concepts to enhance data analysis and improve predictive modeling capabilities. These concepts play a crucial role in extracting valuable insights, identifying patterns, and making accurate predictions based on the dataset. The following ML concepts were utilized in this project:

- **Random Forest Regression:** Random Forest Regression is an ensemble learning method that combines multiple decision trees to create a robust predictive model. It is used for imputing missing values and predicting target variables based on other features. The Random Forest Regression algorithm is capable of handling non-linear relationships and capturing complex interactions within the data.
- **Simple Imputer:** Simple Imputer is a technique used for imputing missing values in the dataset. It replaces the missing values with appropriate substitutes, such as the mean, median, or most frequent value of the corresponding feature. Simple Imputer helps ensure that missing values do not adversely affect the subsequent analysis and modeling tasks.
- **Outlier Detection:** Outlier detection is a technique used to identify and handle anomalous data points that deviate significantly from the majority of the dataset. Outliers can adversely impact the analysis and modeling results, leading to biased or inaccurate predictions. Various outlier detection methods, such as the Z-score method or interquartile range (IQR), can be applied to identify and remove outliers, ensuring the integrity and reliability of the dataset.
- **Feature Scaling:** Feature scaling is a process that brings the features of the dataset to a similar scale or range. It ensures that all features contribute equally during analysis and modeling, avoiding the dominance of certain features due to their larger scales. Common feature scaling techniques include Min-Max scaling and Z-score normalization.

These ML concepts are employed strategically to handle missing values, reduce dimensionality, detect outliers, and normalize features, enabling accurate analysis, modeling, and prediction tasks. By incorporating these concepts into the model, the project aims to improve data quality, enhance the performance of the predictive models, and derive meaningful insights from the dataset.

1.8 Data Visualization:

- Exploratory Data Analysis and Visualization: In this phase, we perform exploratory data analysis (EDA) and utilize data visualization techniques to gain insights and effectively communicate findings. Descriptive statistics, tables, and graphs provide a comprehensive overview of the pre-processed data.
- Data Description and Summary Statistics: To facilitate understanding, we present a detailed data description and summary statistics. The descriptive table offers insights into categories, sub-categories/columns, and frequencies. Additional summary statistics like mean, median, standard deviation, and quartiles provide information about numerical variables. We also assess missing/null values, highlighting potential data gaps and guiding data cleaning. The table displays the count and percentage of missing values and the total count of unique values.
- Subset Data Display: To showcase a representative sample of pre-processed data, we offer a tabular subset display. This table presents relevant columns and values, enabling users to explore the dataset's structure, identify patterns, and gain insights. Pagination and filtering options facilitate efficient navigation and analysis based on user requirements.
- Data Completeness Graph: We use the matplotlib library to create a comprehensive data completeness graph. This intuitive visualization displays the proportion of complete data in each category. Color coding and bar lengths quickly indicate completeness levels, helping identify data gaps and areas requiring further collection or cleaning. Interactive features like tooltips, zooming, and panning enhance user experience and detailed exploration.

Through these visualization techniques, users can delve into the pre-processed data, uncover hidden patterns, and make informed decisions. Descriptive statistics, subset data display, and the data completeness graph provide a holistic view, serving as a foundation for further analysis and exploration.

1.9 Empowering Users through Interactive Data Visualization and Informed Data Selection:

Data visualization plays a crucial role in our project as it enables users to interactively explore and select relevant data categories for processing. By providing visual representations of the data, we empower users to gain a comprehensive understanding of the dataset's structure, patterns, and characteristics. Through descriptive tables, summary statistics, and visually appealing graphs, users can identify key insights, trends, and outliers within the data. This visual exploration facilitates informed decision-making by allowing users to selectively choose the categories that are most relevant to their specific needs and objectives. By interacting with the visualizations, users can assess the completeness, distribution, and relationships within the data, guiding them in making informed choices regarding the final selection of categories for processing. Ultimately, data visualization serves as a powerful tool that enhances user engagement, facilitates data exploration, and enables users to unlock the true potential of the dataset for their specific requirements.

Part 2: **WEBSITE Design and APIs Writing**

2.1.1 Introduction:

In our project, we have combined the power of Sanic APIs with an intuitive website design to create a seamless data preprocessing solution. Sanic, a high-performance Python web framework, handles asynchronous requests efficiently, enabling quick and responsive data transformations. Our user-friendly website provides an intuitive interface for users to interact with our data preprocessing model, submit datasets, define preprocessing operations, and retrieve processed data effortlessly.

By integrating Sanic APIs and our website design, we offer several advantages. Sanic's asynchronous processing capabilities allow for parallel execution of preprocessing operations, resulting in faster response times. The website design prioritizes simplicity, interactivity, and visual appeal, ensuring a pleasant and engaging user experience.

Our combined approach creates a collaborative environment where users can experiment with different preprocessing options and gain valuable insights from their data. The integration of Sanic APIs and the website design enables us to provide a comprehensive platform that caters to the diverse needs of data scientists and analysts. It enhances their productivity and facilitates insightful data preprocessing.

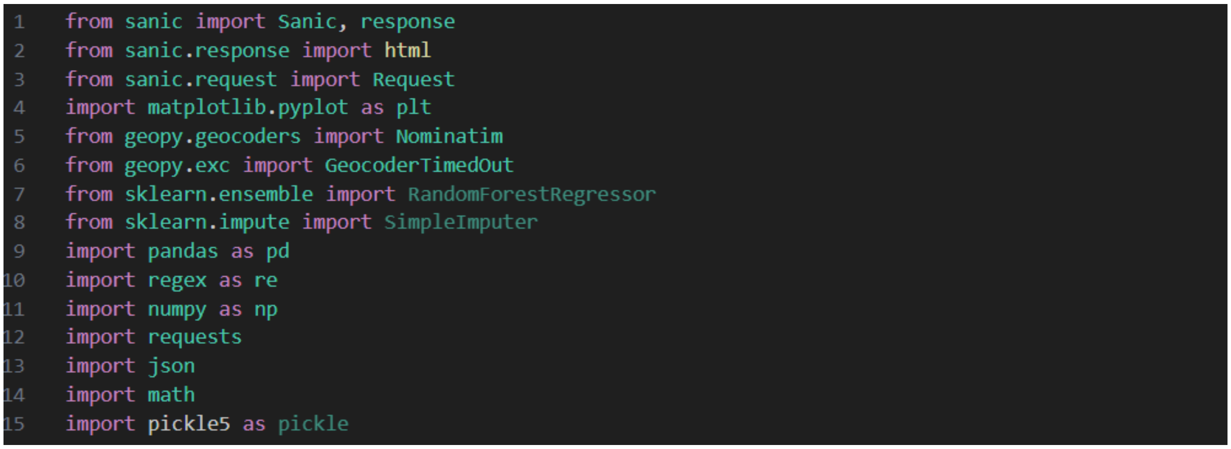
overall, our combined approach of Sanic APIs and website design forms the foundation of our efficient data preprocessing solution. The integration ensures optimal performance, responsiveness, and user-friendliness, allowing users to seamlessly interact with our model and preprocess their data effortlessly.

2.1.2 Why Sanic Framework for APIs?

In our project, the choice of using the Sanic framework for API design is driven by several key reasons:

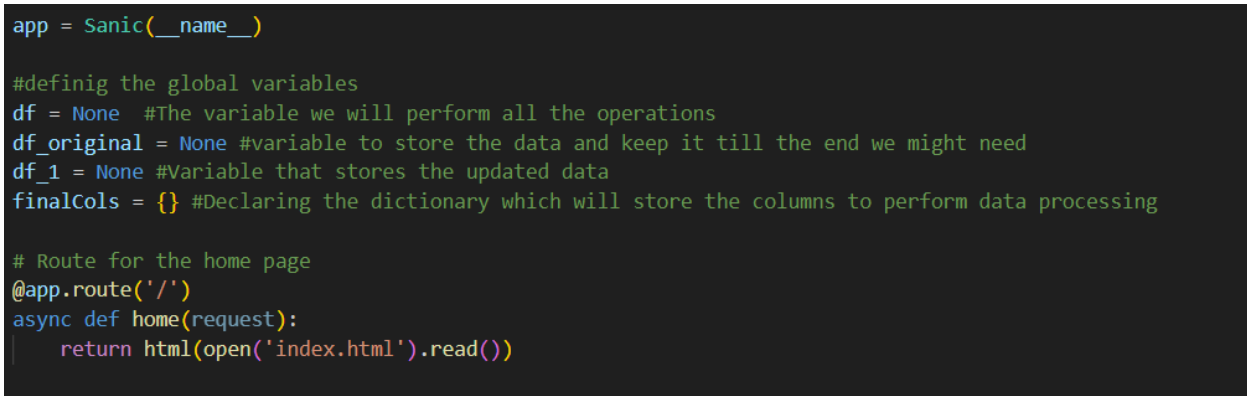
- High Performance: Sanic excels in performance and asynchronous capabilities, making it an ideal choice for fast and responsive APIs. Its ability to handle numerous concurrent requests efficiently ensures smooth and effective data processing.
- Scalability: Sanic's architecture is designed for handling high loads and effortless scalability. By utilizing asyncio, a Python library for asynchronous programming, Sanic can handle multiple requests concurrently. This ensures seamless scaling as the user base expands or when working with large datasets.
- Pythonic Approach: Sanic adheres to Python's philosophy of readability and simplicity. It embraces Pythonic coding practices, empowering developers to write clean and maintainable code. This aligns perfectly with our project's objective of developing a user-friendly and accessible API.

2.1.3 Packages installed:



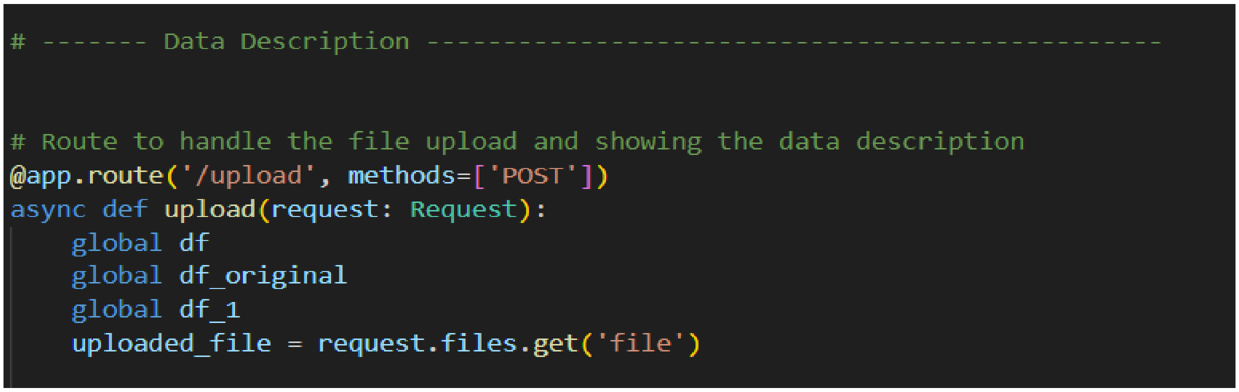
```python
from sanic import Sanic, response
from sanic.response import html
from sanic.request import Request
import matplotlib.pyplot as plt
from geopy.geocoders import Nominatim
from geopy.exc import GeocoderTimedOut
from sklearn.ensemble import RandomForestRegressor
from sklearn.impute import SimpleImputer
import pandas as pd
import regex as re
import numpy as np
import requests
import json
import math
import pickle5 as pickle
```

2.1.4 Defining API instance, global variables, and a route to the file upload page:



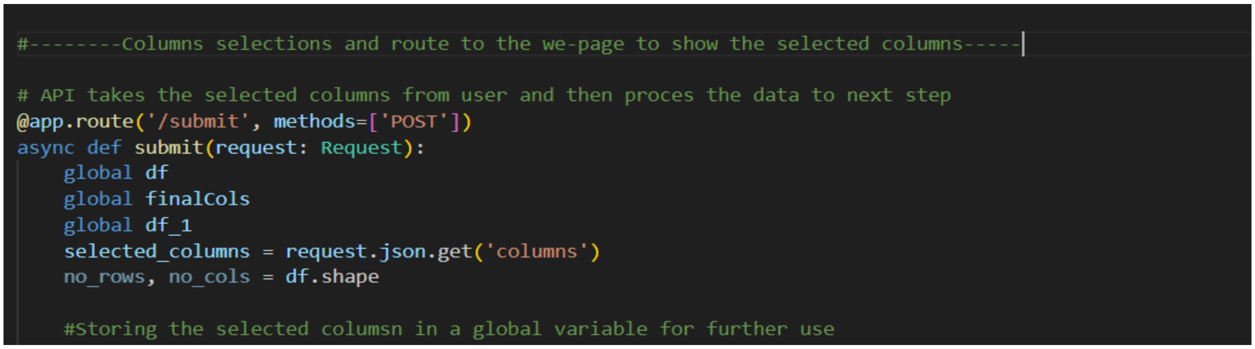
```python
app = Sanic(__name__)

#definig the global variables
df = None  #The variable we will perform all the operations
df_original = None #variable to store the data and keep it till the end we might need
df_1 = None #Variable that stores the updated data
finalCols = {} #Declaring the dictionary which will store the columns to perform data processing

# Route for the home page
@app.route('/')
async def home(request):
    return html(open('index.html').read())
```

2.1.5 Upload Api - it deals with retrieving the data file from server, converting it into a pandas data frame and then sending the data description details to the ser



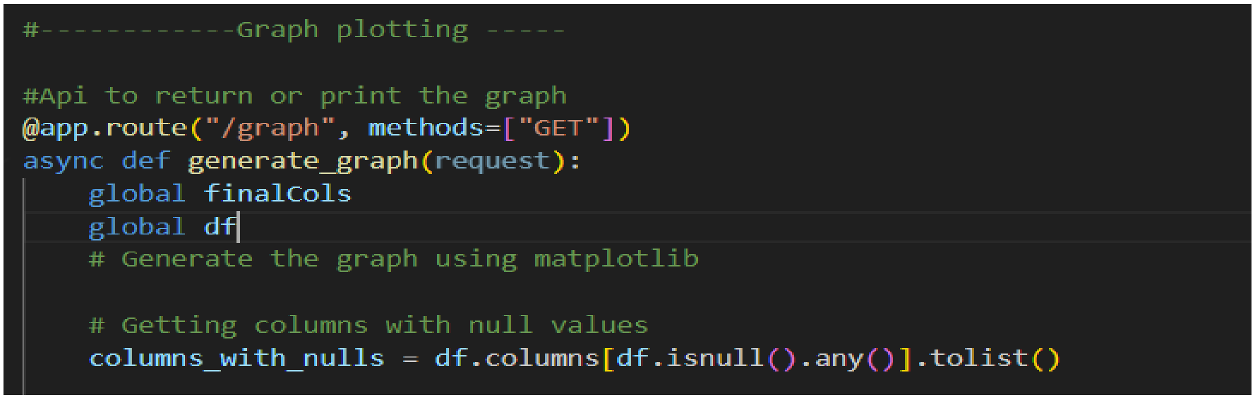
```python
# ------- Data Description ---------------------------------------------


# Route to handle the file upload and showing the data description
@app.route('/upload', methods=['POST'])
async def upload(request: Request):
    global df
    global df_original
    global df_1
    uploaded_file = request.files.get('file')
```

2.1.6 Asking the user to select the final columns after providing data description, it stores the selected columns and sends them back to show the user(these columns has been selected):



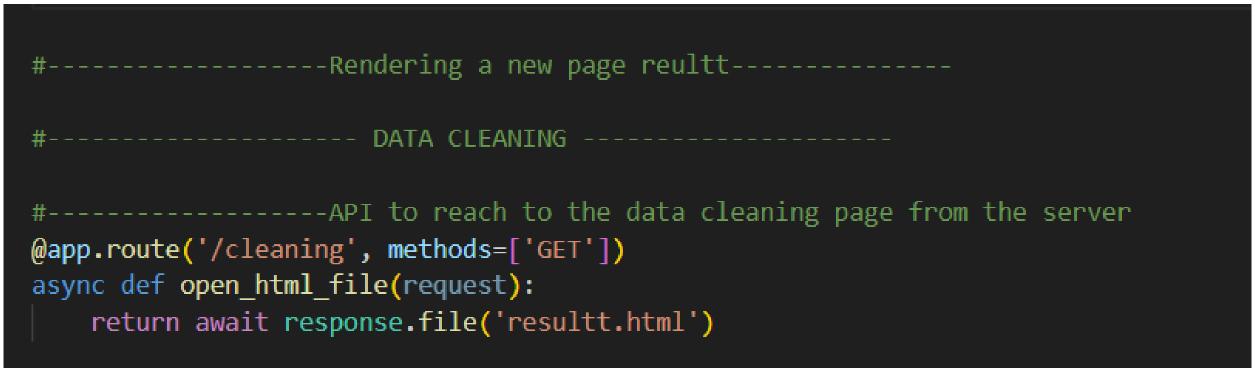
```
#--------Columns selections and route to the we-page to show the selected columns-----

# API takes the selected columns from user and then proces the data to next step
@app.route('/submit', methods=['POST'])
async def submit(request: Request):
    global df
    global finalCols
    global df_1
    selected_columns = request.json.get('columns')
    no_rows, no_cols = df.shape

    #Storing the selected columsn in a global variable for further use
```

2.1.7 Plotting the graph and sending it to the web-page



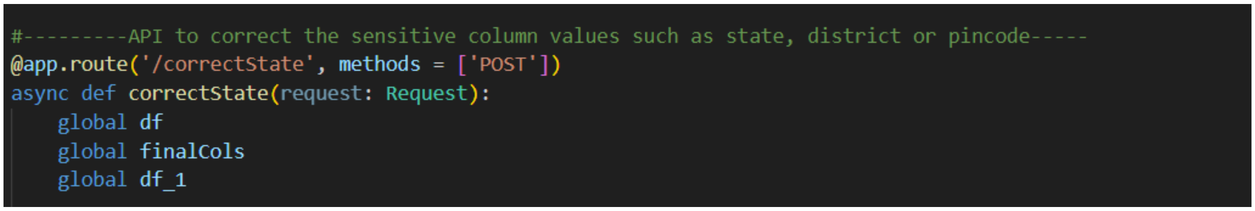
```
#------------Graph plotting -----

#Api to return or print the graph
@app.route("/graph", methods=["GET"])
async def generate_graph(request):
    global finalCols
    global df
    # Generate the graph using matplotlib

    # Getting columns with null values
    columns_with_nulls = df.columns[df.isnull().any()].tolist()
```

2.1.8 Rendering the new page to show the data cleaning results to the user and interact with the user



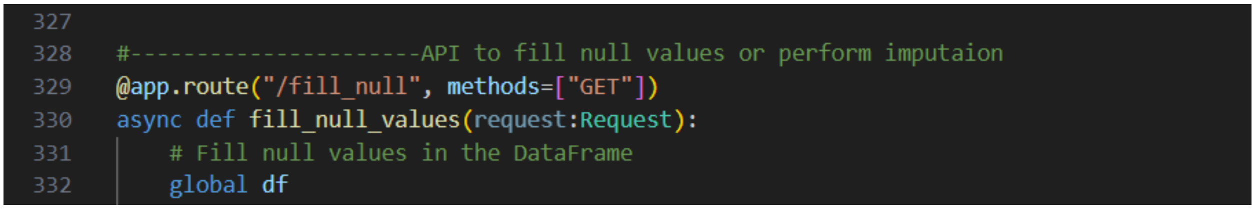
```
#------------------Rendering a new page reultt---------------

#-------------------- DATA CLEANING --------------------

#------------------API to reach to the data cleaning page from the server
@app.route('/cleaning', methods=['GET'])
async def open_html_file(request):
    return await response.file('resultt.html')
```

Removing abnormal text, and errors, reshaping and reformatting the address, dealing with incorrect values, outliers and replacing all of these by NaNs:



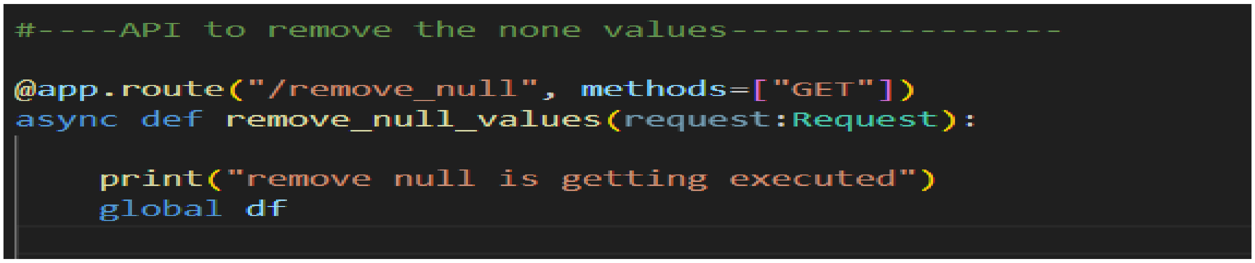
```
#---------API to correct the sensitive column values such as state, district or pincode-----
@app.route('/correctState', methods = ['POST'])
async def correctState(request: Request):
    global df
    global finalCols
    global df_1
```

2.1.9 Filling the none values present in the data, it includes imputation for sensitive categories and general categories, here we use ML models:



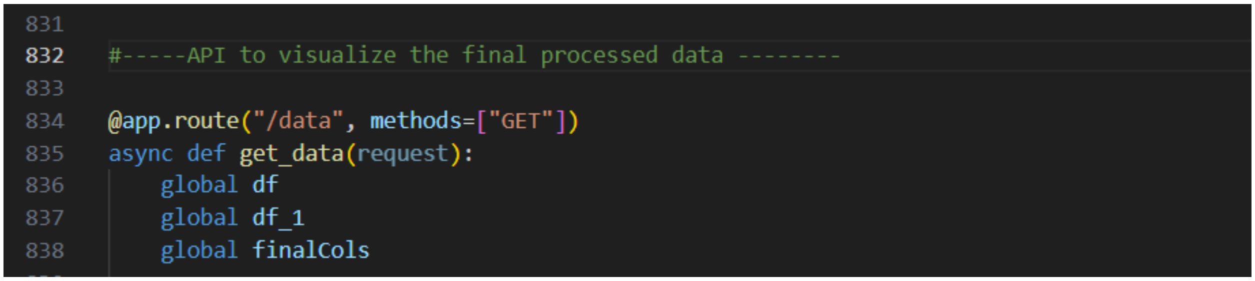
```

#---------------------API to fill null values or perform imputaion
@app.route("/fill_null", methods=["GET"])
async def fill_null_values(request:Request):
    # Fill null values in the DataFrame
    global df
```

2.1.10 If the user doesn't want to fill the null values using our models and wants to filter the complete columns:



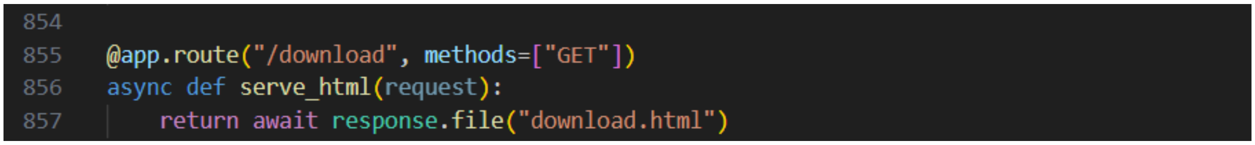
```
#----API to remove the none values----------------

@app.route("/remove_null", methods=["GET"])
async def remove_null_values(request:Request):

    print("remove null is getting executed")
    global df
```

2.1.11 API that sends the final processed data to the html page and user see it in a table format:



```

#-----API to visualize the final processed data --------

@app.route("/data", methods=["GET"])
async def get_data(request):
    global df
    global df_1
    global finalCols
```

2.1.12 API that enables the user to download the final pre-processed data for their use:

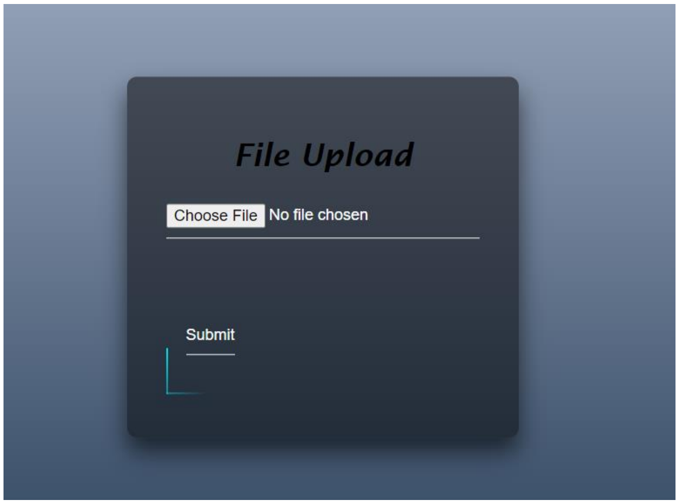


```

@app.route("/download", methods=["GET"])
async def serve_html(request):
    return await response.file("download.html")
```

2.2 Website Design for the user to interact with APIs in data preprocessing

2.2.1 1st Page: Data file upload

The Upload Page provides users with a simple and intuitive interface to upload their data files. Users can easily select and upload various file formats, such as **Excel, CSV, and text files**. The page features an interactive upload **button with a visually engaging hover effect**, adding a touch of interactivity to the upload process. This user-friendly design ensures a seamless transition from the user's local machine to the subsequent data processing steps.



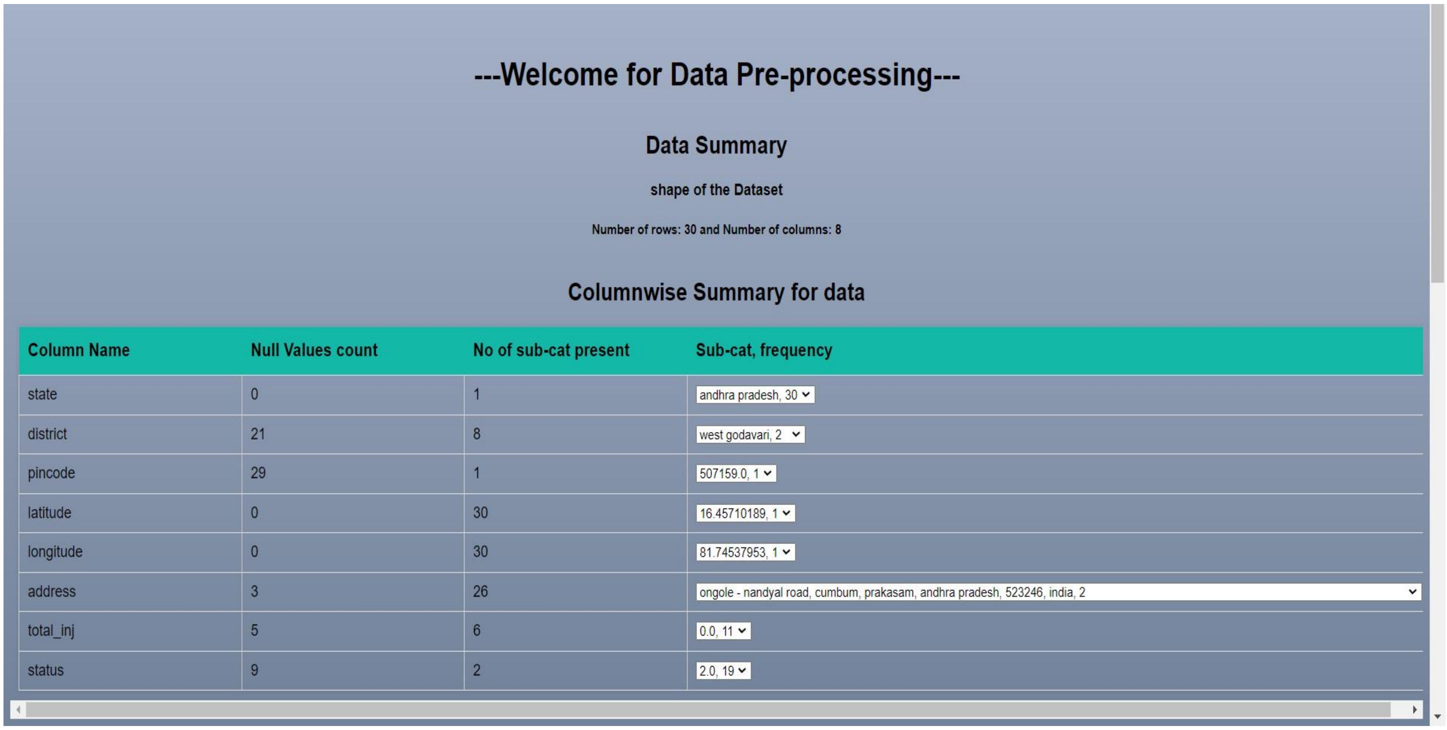
2.2.2 2nd Page: Providing the Data Description and Visualization

The Data Overview Page provides users with a comprehensive view of the uploaded dataset. It presents a detailed table showcasing the categories and subcategories within each column, along with their respective frequencies. Additionally, the table highlights the number of null values present in each column, enabling users to identify areas that may require data cleaning or imputation.

Beyond the table, the page offers a visually informative Data Completeness Graph. This graph visually represents the completeness of the dataset, allowing users to gauge the overall data quality. By analyzing the graph, users can identify columns with high completeness and prioritize them for further analysis or processing. This feature facilitates informed decision-making by assisting users in selecting the most suitable columns or categories for their specific data preprocessing requirements.

The combination of the descriptive table and the Data Completeness Graph provides users with a holistic understanding of the dataset's structure and completeness. This knowledge empowers users to make informed choices regarding the columns or categories they wish to include in the subsequent data preprocessing steps. The Data Overview Page serves as a valuable starting point for users, enabling them to navigate and explore the dataset effectively.

2.2.3 Showing Data Description to the user



---Welcome for Data Pre-processing---

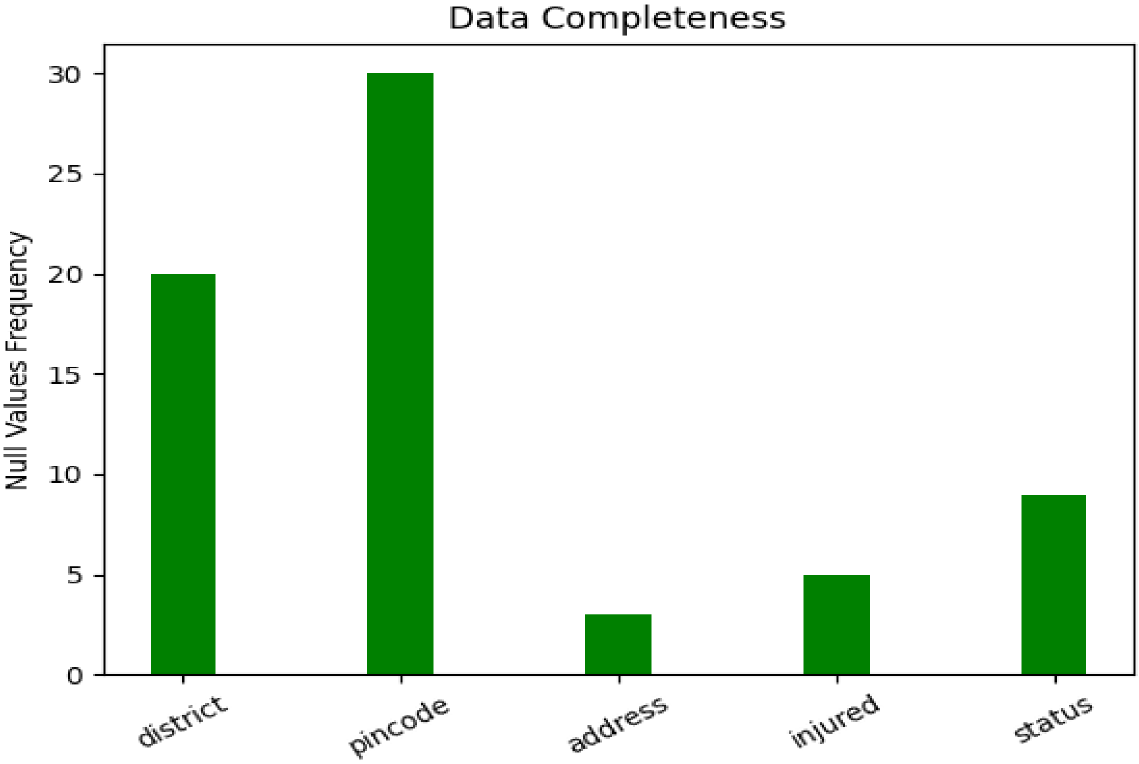
Data Summary

shape of the Dataset

Number of rows: 30 and Number of columns: 8

Columnwise Summary for data

Column Name	Null Values count	No of sub-cat present	Sub-cat, frequency
state	0	1	andhra pradesh, 30
district	21	8	west godavari, 2
pincode	29	1	507159.0, 1
latitude	0	30	16.45710189, 1
longitude	0	30	81.74537953, 1
address	3	26	ongole - nandyal road, cumbum, prakasam, andhra pradesh, 523246, india, 2
total_inj	5	6	0.0, 11
status	9	2	2.0, 19



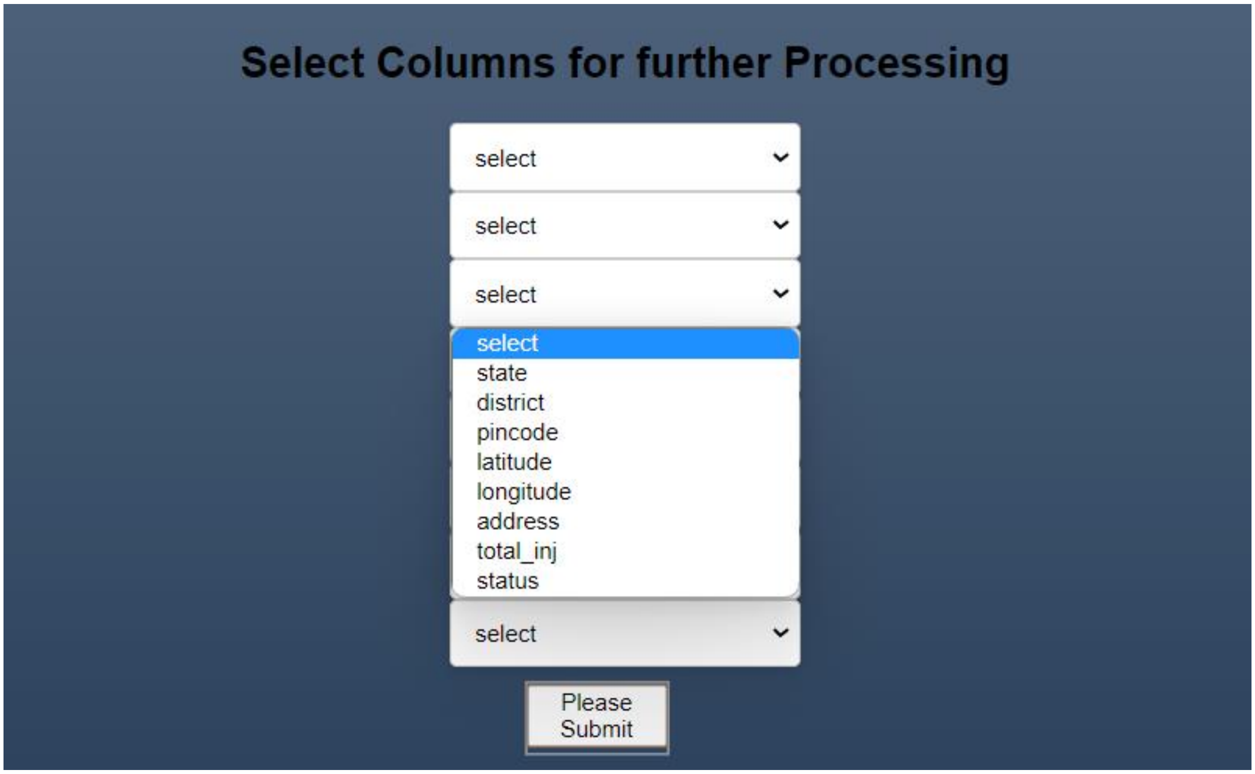
2.2.4 3rd Page: Feature selection and proceeding for further processing of data

The Column Selection Page is designed to provide users with flexibility in choosing the specific columns they want to include in the data preprocessing step. The page presents a set of dropdown menus, each corresponding to a column in the uploaded dataset.

Each dropdown menu contains the names of the columns as options, allowing users to select the desired columns for further processing. Users have the freedom to choose any number of columns by selecting them from the respective dropdown menus. If a user wants to exclude a column from the preprocessing step, they can simply leave the corresponding dropdown menu unselected.

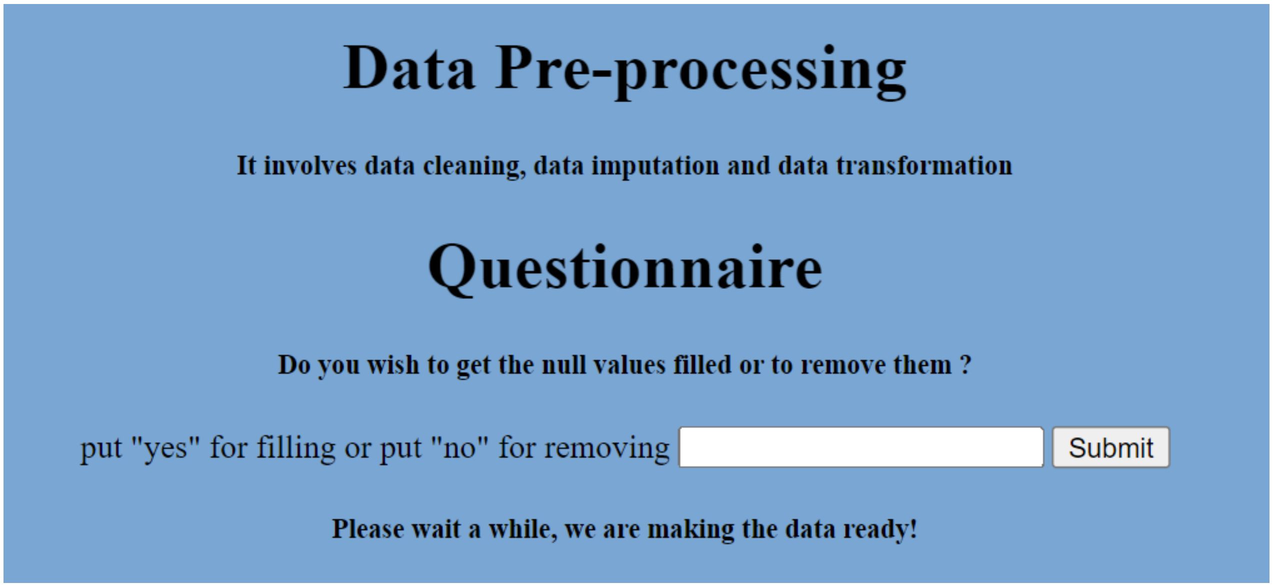
To ensure data integrity and prevent duplication, the system is designed to handle cases where the user selects the same column in multiple dropdown menus. In such instances, the column is considered only once in the final selection process. This mechanism prevents redundant or erroneous data processing and ensures accurate results.

Upon selecting the desired columns using the dropdown menus, users can proceed by clicking the submission button. The selected column choices are then sent to the backend API, where the final columns for data preprocessing are determined based on the user's selections. This seamless integration between the frontend interface and the backend API ensures a smooth and efficient data preprocessing workflow.



2.2.5 4th Page: Dealing with "None" values in the data after the data cleaning step:

The Null Value Handling Page provides users with options to handle null values in the dataset. They can choose to either remove null values or fill them. Removing null values allows users to work with the original dataset as-is, without any modifications. On the other hand, filling null values involves using advanced techniques to replace missing values with accurate and appropriate values. After handling null values, the page presents the final preprocessed data in a table format for visualization and inspection. Users can also download the processed dataset in CSV format for further analysis and integration with other tools. This page enables users to make informed decisions about null value treatment, enhancing the quality and reliability of their data.



Data Pre-processing

It involves data cleaning, data imputation and data transformation

Questionnaire

Do you wish to get the null values filled or to remove them ?

put "yes" for filling or put "no" for removing

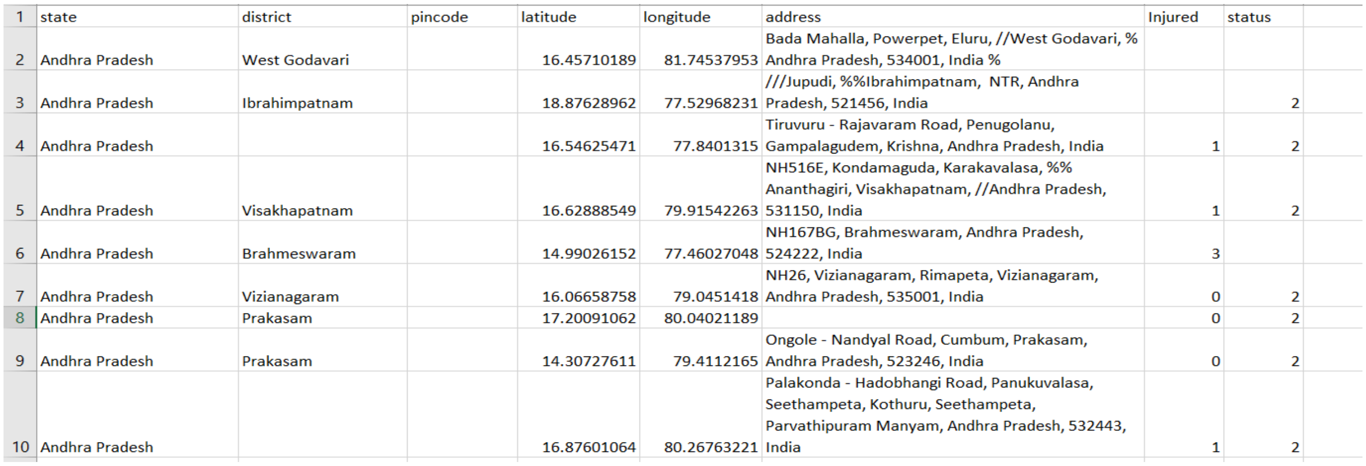
Please wait a while, we are making the data ready!

2.2.6 5th Page: Visualizing the Pre-processed data and downloading it:

The Final Processed Data Page provides users with access to the processed data generated by the API after performing various operations such as data cleaning, transformation, feature selection, and imputation. This page displays a subset of the processed data in a table format, allowing users to get an overview of how the final data looks. The table includes all the selected columns and their corresponding values. Additionally, a download button is provided, enabling users to download the complete processed data in CSV format. This allows users to further analyze and explore the data using their preferred tools and techniques. The Final Processed Data Page serves as the endpoint of the data processing pipeline, providing users with the output they need for their data-driven tasks and decision-making processes.

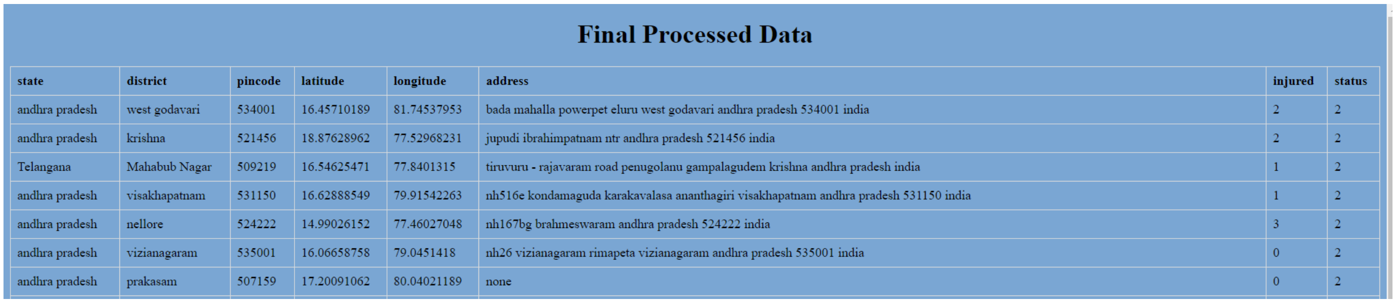
2.3 Comparison between input data and pre-processed data:

Before:



1	state	district	pincode	latitude	longitude	address	Injured	status
2	Andhra Pradesh	West Godavari		16.45710189	81.74537953	Bada Mahalla, Powerpet, Eluru, //West Godavari, % Andhra Pradesh, 534001, India %		
3	Andhra Pradesh	Ibrahimpatnam		18.87628962	77.52968231	///Jupudi, %%Ibrahimpatnam, NTR, Andhra Pradesh, 521456, India		2
4	Andhra Pradesh			16.54625471	77.8401315	Tiruvuru - Rajavaram Road, Penugolanu, Gampalagudem, Krishna, Andhra Pradesh, India	1	2
5	Andhra Pradesh	Visakhapatnam		16.62888549	79.91542263	NH516E, Kondamaguda, Karakavalasa, %% Ananthagiri, Visakhapatnam, //Andhra Pradesh, 531150, India	1	2
6	Andhra Pradesh	Brahmeswaram		14.99026152	77.46027048	NH167BG, Brahmeswaram, Andhra Pradesh, 524222, India	3	
7	Andhra Pradesh	Vizianagaram		16.06658758	79.0451418	NH26, Vizianagaram, Rimapeta, Vizianagaram, Andhra Pradesh, 535001, India	0	2
8	Andhra Pradesh	Prakasam		17.20091062	80.04021189		0	2
9	Andhra Pradesh	Prakasam		14.30727611	79.4112165	Ongole - Nandyal Road, Cumbum, Prakasam, Andhra Pradesh, 523246, India	0	2
10	Andhra Pradesh			16.87601064	80.26763221	Palakonda - Hadobhangi Road, Panukuvalasa, Seethampeta, Kothuru, Seethampeta, Parvathipuram Manyam, Andhra Pradesh, 532443, India	1	2

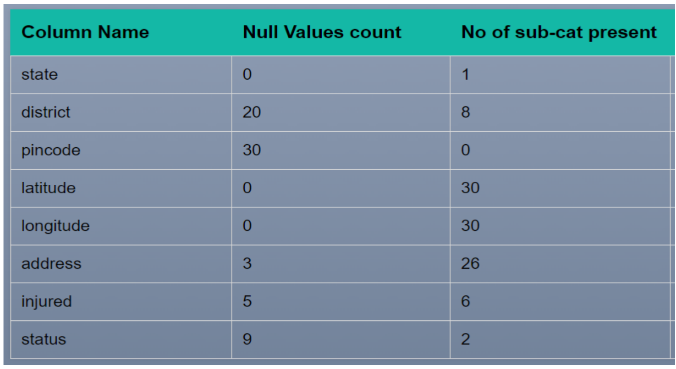
After:



Final Processed Data

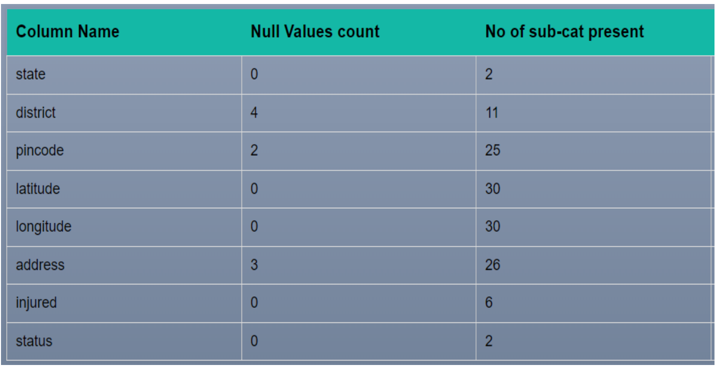
state	district	pincode	latitude	longitude	address	injured	status
andhra pradesh	west godavari	534001	16.45710189	81.74537953	bada mahalla powerpet eluru west godavari andhra pradesh 534001 india	2	2
andhra pradesh	krishna	521456	18.87628962	77.52968231	jupudi ibrahimpatnam ntr andhra pradesh 521456 india	2	2
Telangana	Mahabub Nagar	509219	16.54625471	77.8401315	tiruvuru - rajavaram road penugolanu gampalagudem krishna andhra pradesh india	1	2
andhra pradesh	visakhapatnam	531150	16.62888549	79.91542263	nh516e kondamaguda karakavalasa ananthagiri visakhapatnam andhra pradesh 531150 india	1	2
andhra pradesh	nellore	524222	14.99026152	77.46027048	nh167bg brahmeswaram andhra pradesh 524222 india	3	2
andhra pradesh	vizianagaram	535001	16.06658758	79.0451418	nh26 vizianagaram rimapeta vizianagaram andhra pradesh 535001 india	0	2
andhra pradesh	prakasam	507159	17.20091062	80.04021189	none	0	2

Before Pre-processing

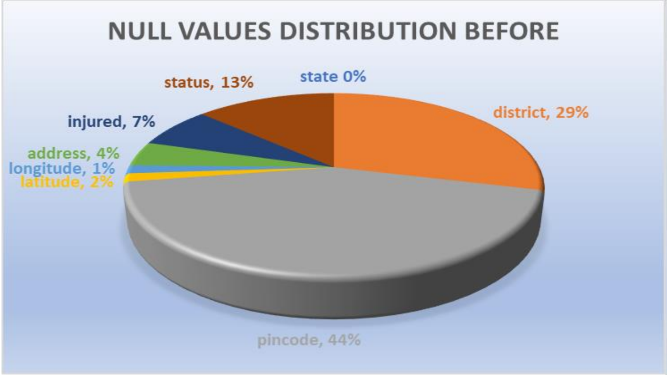


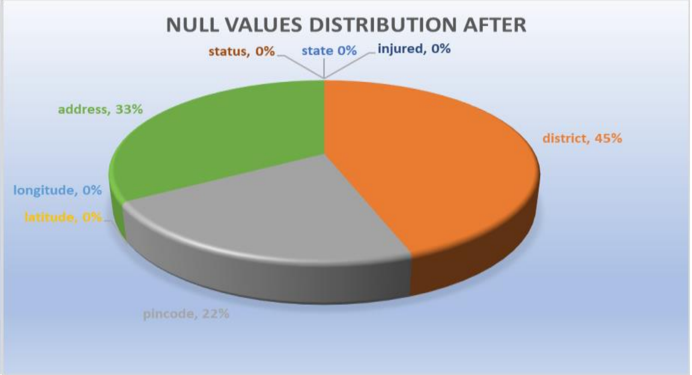
Column Name	Null Values count	No of sub-cat present
state	0	1
district	20	8
pincode	30	0
latitude	0	30
longitude	0	30
address	3	26
injured	5	6
status	9	2

After Pre-processing



Column Name	Null Values count	No of sub-cat present
state	0	2
district	4	11
pincode	2	25
latitude	0	30
longitude	0	30
address	3	26
injured	0	6
status	0	2





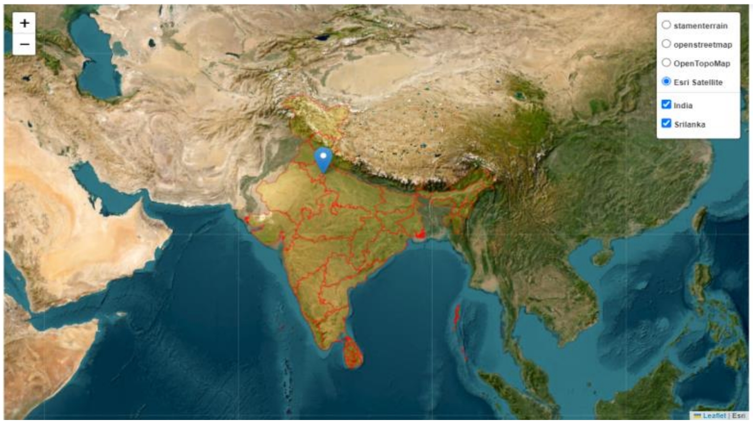
Part 3: Pre-processed data use case

3.1 Accidental Sites and hotspots generation:

The graph created using the Folium library is a powerful tool for visualizing the geographical distribution of accidents and extracting valuable insights from the processed data. By plotting the accident coordinates on a map, it provides a comprehensive understanding of accident-prone areas and their distribution patterns, highlighting the need for targeted interventions. One key functionality of the graph is the identification of hotspots, which are areas with a higher frequency of accidents. By analyzing the density and clustering of accidents, the graph enables policymakers and law enforcement agencies to prioritize safety measures and allocate resources strategically. This information is crucial for designing effective interventions and implementing strategies to address the specific needs of accident-prone areas. The graph also incorporates interactive pop-ups that provide detailed information about each accident location. Users can explore the accidents in greater depth by interacting with specific markers or hotspots, gaining insights into the contributing factors and facilitating data-driven decision-making.

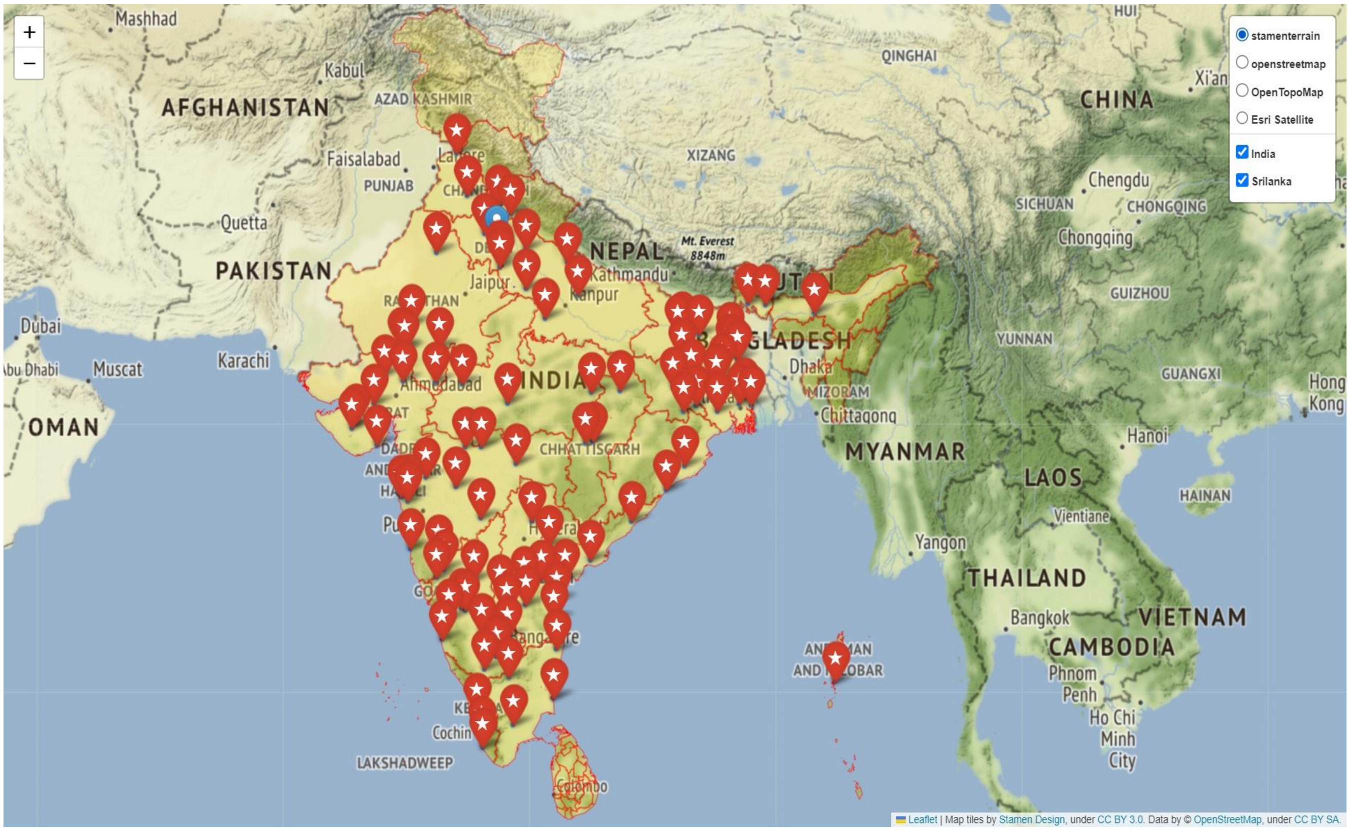
Additionally, the graph offers different map layers, including street maps, terrain maps, and satellite imagery. This multi-layered approach enhances the interpretability of the data, providing diverse perspectives on the accident sites and their surroundings. Users can switch between layers to gain valuable insights into environmental factors and infrastructural attributes that may influence accident occurrences, enabling a holistic understanding of accident patterns.

The importance of this use case lies in its ability to present complex information in a visually compelling and intuitive manner. By leveraging geospatial visualization of the processed data, it effectively communicates the distribution, concentration, and characteristics of accidents. This empowers decision-makers, stakeholders, and community members to make informed choices regarding safety interventions, policy implementation, and resource allocation.



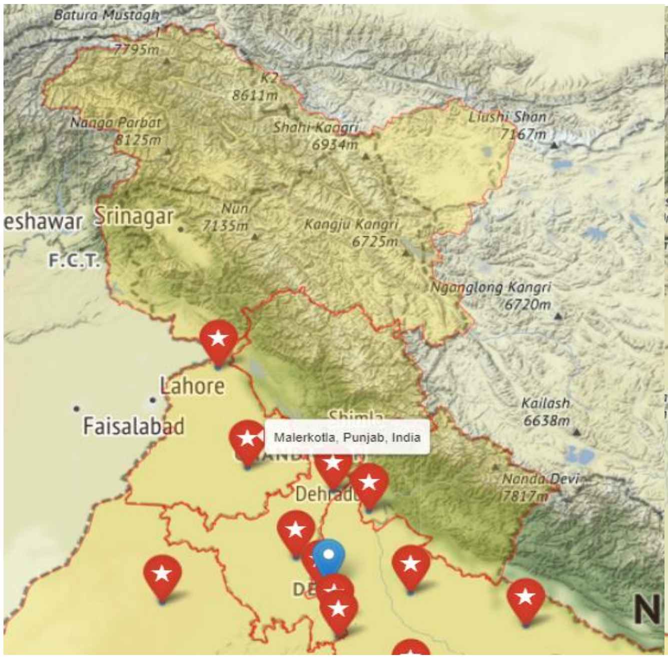
3.2 Indian map having accidental locations:

The map contains different layers for the map visualization that are shown in the right and the locations mapped on the map. It shows the location name and image of the accidental site in a pop-up.

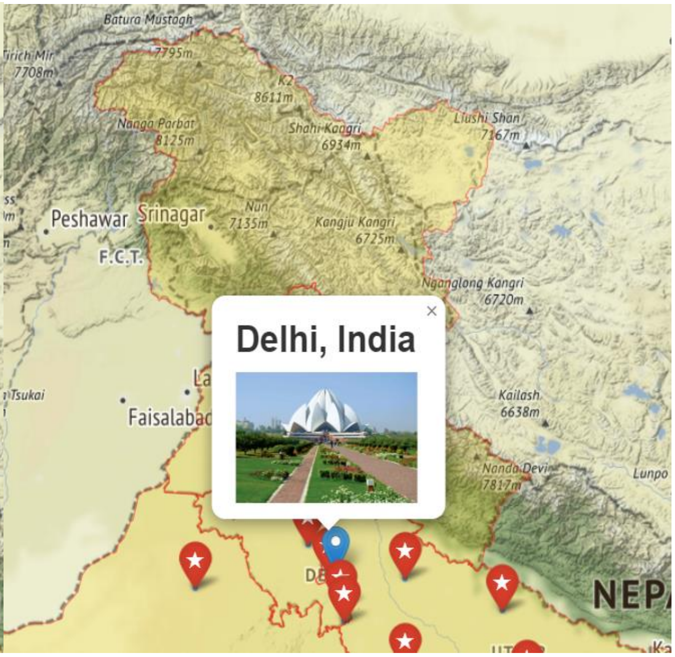


● stamenterrain
○ openstreetmap
○ OpenTopoMap
○ Esri Satellite
☑ India
☑ Srilanka

Leaflet | Map tiles by Stamen Design, under CC BY 3.0. Data by © OpenStreetMap, under CC BY SA.



Malerkotla, Punjab, India



Delhi, India

4. Challenges Faced:

During the development of this project, we encountered several challenges that required careful consideration and innovative solutions. The following are the key challenges we faced:

1. **Data Heterogeneity:** Dealing with different types of data formats (e.g., Excel, CSV, text) and varying data structures posed a significant challenge. We had to develop robust algorithms and parsers to handle these diverse data sources and ensure seamless data ingestion and processing.
2. **Scalability:** As the project aimed to support large datasets, scalability was a crucial consideration. Optimizing the data preprocessing pipeline to handle large volumes of data efficiently and without compromising performance required careful design choices and algorithmic optimizations.
3. **Null Value Handling:** Dealing with missing values is a common challenge in data preprocessing. Developing a robust strategy to handle null values based on user preferences while ensuring data integrity and preserving statistical properties required careful consideration and testing.
4. **Algorithm Selection:** Choosing the most suitable algorithms and techniques for data cleaning, transformation, feature selection, and imputation was a critical decision. We had to carefully evaluate various algorithms, considering factors such as performance, accuracy, and compatibility with different data types, to ensure the best possible results.
5. **Performance Optimization:** Achieving a balance between computational efficiency and result accuracy was a constant challenge. We continually optimized our algorithms and code to minimize processing time while maintaining high-quality data preprocessing outcomes.
6. **Integration of APIs:** Integrating external APIs for specific data processing tasks required thorough understanding and coordination. We had to ensure seamless integration, data compatibility, and secure communication between our application and the APIs.

Overcoming these challenges required a combination of technical expertise, thorough research, iterative development, and effective communication within the project team. By addressing these challenges, we were able to develop a robust and user-friendly data preprocessing solution that meets the needs of data scientists and analysts.

5. Conclusion:

This project demonstrates the power of combining machine learning algorithms, data processing techniques, and web-based APIs to create an efficient and user-friendly data preprocessing solution. By addressing common data preprocessing challenges and providing interactive visualization tools, the project empowers users to transform raw and messy data into clean and actionable insights. The streamlined workflow, customizable features, and focus on user experience make this project a valuable asset for data scientists, analysts, and researchers working with diverse datasets. Ultimately, this project contributes to enhancing data quality, enabling more accurate and reliable analyses, and driving informed decision-making in various domains.

------------------Thanks------------------