AMITY UNIVERSITY
KOLKATA

PROJECT REPORT

Name: Utsav Sarkar
Enrolment No: A910119821003
Stream: Artificial Intelligence 2021-2025

Name: Ankur Kuntal Chakraborty
Enrolment No: A91005221041
Stream: Computer Science (B) 2021-2025

Institute: Amity School of Engineering & Technology

Course Title: In-House Practical Training
Course Code: ETPT100
Semester: 5
Submitted to: Sudip Chatterjee

Project Report: Accident Zone Prediction and Visualization

Introduction

This project focuses on predicting accident zones and visualizing the predictions on a map. The goal is to provide insights into potential accident-prone areas based on historical data and machine learning models.

Libraries Used

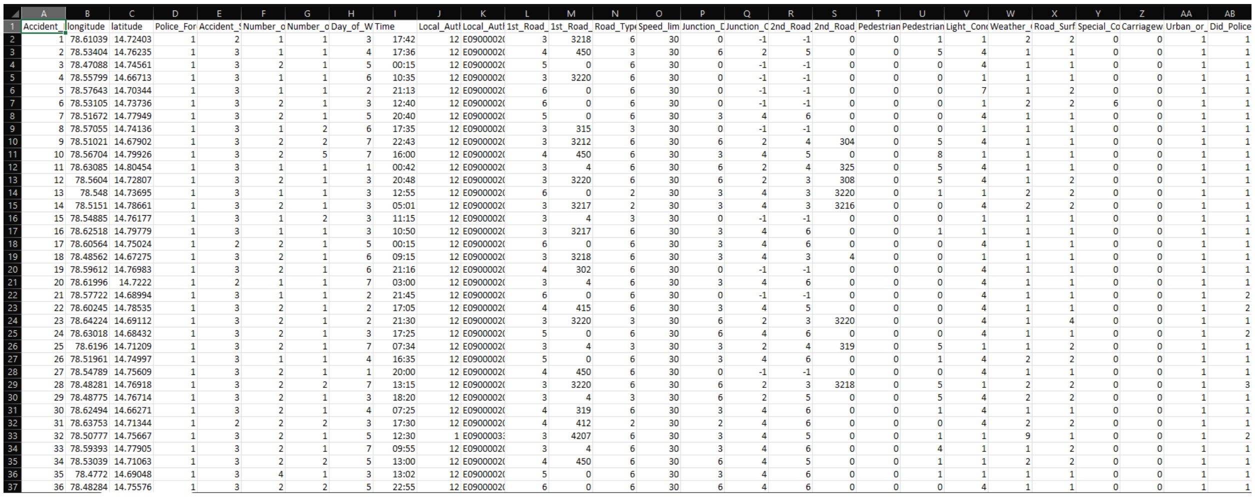
The project utilizes the following Python libraries:

- 'pandas' for data manipulation and analysis.
- 'scikit-learn' for machine learning model creation.
- 'folium' for creating interactive maps.

Data

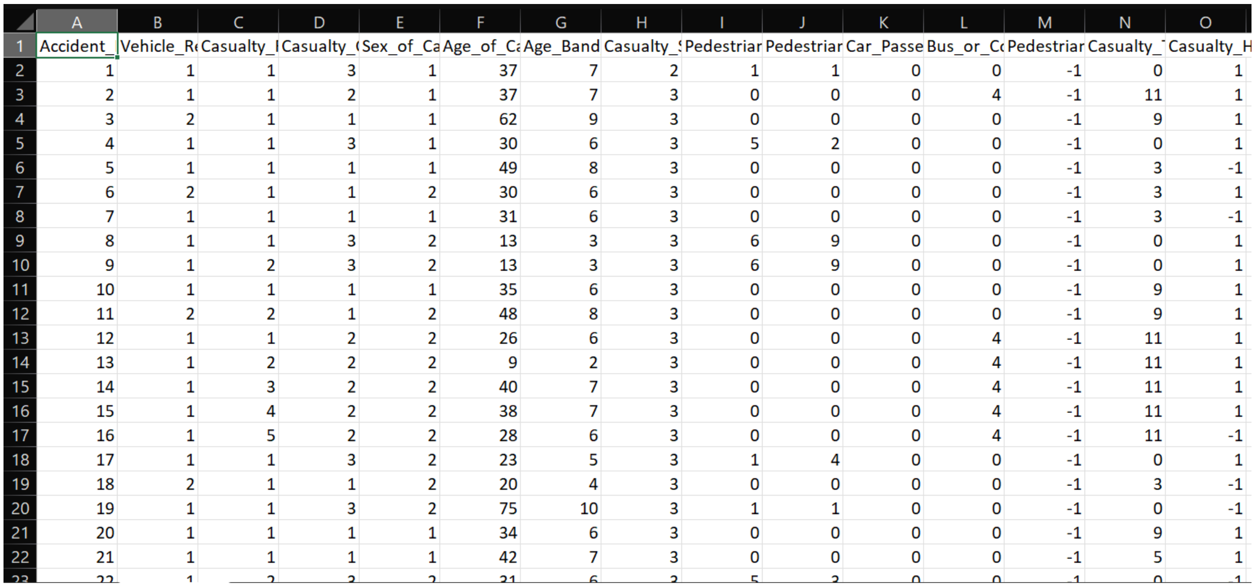
The project involves three main datasets:

1. Accidents Data (`AccidentsBig.csv`): Contains information about accidents, including location coordinates (latitude, longitude), severity, and other relevant details.



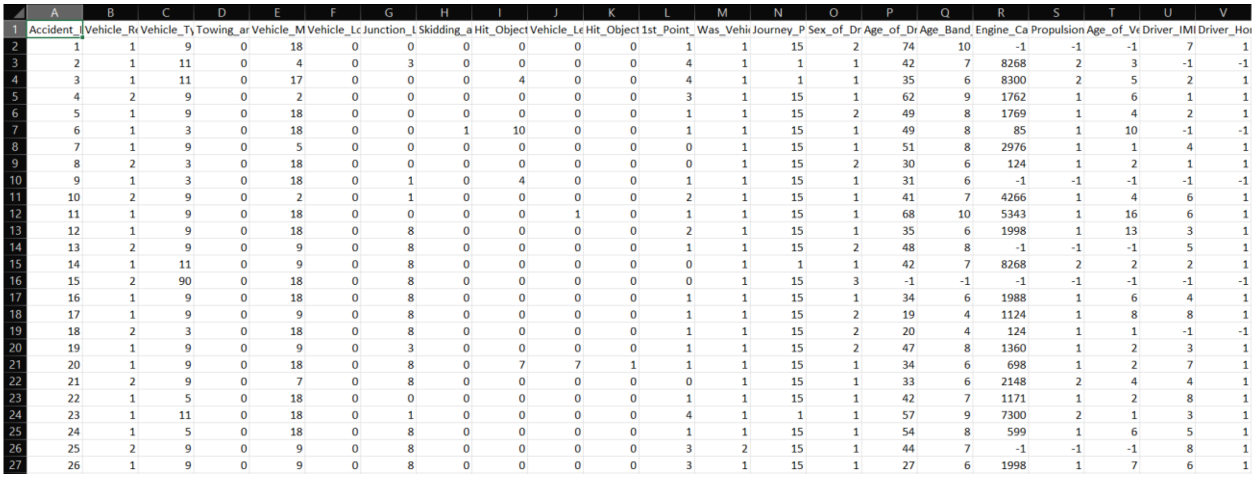
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	Accident_I	longitude	latitude	Police_For	Accident_S	Number_o	Number_o	Day_of_W	Time	Local_Autl	Local_Autl	1st_Road_	1st_Road_	Road_Type	Speed_lim	Junction_D	Junction_C	2nd_Road_	2nd_Road_	Pedestrian	Pedestrian	Light_Cond	Weather_(	Road_Surf	Special_Co	Carriagew	Urban_or_	Did_Police
2	1	78.61039	14.72403	1	2	1	1	3	17:42	12	E0900002(	3	3218	6	30	0	-1	-1	0	0	1	1	2	2	0	0	1	1
3	2	78.53404	14.76235	1	3	1	1	4	17:36	12	E0900002(	4	450	3	30	6	2	5	0	0	5	4	1	1	0	0	1	1
4	3	78.47088	14.74561	1	3	2	1	5	00:15	12	E0900002(	5	0	6	30	0	-1	-1	0	0	0	4	1	1	0	0	1	1
5	4	78.55799	14.66713	1	3	1	1	6	10:35	12	E0900002(	3	3220	6	30	0	-1	-1	0	0	0	1	1	1	0	0	1	1
6	5	78.57643	14.70344	1	3	1	1	2	21:13	12	E0900002(	6	0	6	30	0	-1	-1	0	0	0	7	1	2	0	0	1	1
7	6	78.53105	14.73736	1	3	2	1	3	12:40	12	E0900002(	6	0	6	30	0	-1	-1	0	0	0	1	2	2	6	0	1	1
8	7	78.51672	14.77949	1	3	2	1	5	20:40	12	E0900002(	5	0	6	30	3	4	6	0	0	0	4	1	1	0	0	1	1
9	8	78.57055	14.74136	1	3	1	2	6	17:35	12	E0900002(	3	315	3	30	0	-1	-1	0	0	0	1	1	1	0	0	1	1
10	9	78.51021	14.67902	1	3	2	2	7	22:43	12	E0900002(	3	3212	6	30	6	2	4	304	0	5	4	1	1	0	0	1	1
11	10	78.56704	14.79926	1	3	2	5	7	16:00	12	E0900002(	4	450	6	30	3	4	5	0	0	8	1	1	1	0	0	1	1
12	11	78.63085	14.80454	1	3	1	1	1	00:42	12	E0900002(	3	4	6	30	6	2	4	325	0	5	4	1	1	0	0	1	1
13	12	78.5604	14.72807	1	3	2	1	3	20:48	12	E0900002(	3	3220	6	30	6	2	3	308	0	5	4	1	2	0	0	1	1
14	13	78.548	14.73695	1	3	1	1	3	12:55	12	E0900002(	6	0	2	30	3	4	3	3220	0	1	1	2	2	0	0	1	1
15	14	78.5151	14.78661	1	3	2	1	3	05:01	12	E0900002(	3	3217	2	30	3	4	3	3216	0	0	4	2	2	0	0	1	1
16	15	78.54885	14.76177	1	3	1	2	3	11:15	12	E0900002(	3	4	3	30	0	-1	-1	0	0	0	1	1	1	0	0	1	1
17	16	78.62518	14.79779	1	3	1	1	3	10:50	12	E0900002(	3	3217	6	30	3	4	6	0	0	1	1	1	1	0	0	1	1
18	17	78.60564	14.75024	1	2	2	1	5	00:15	12	E0900002(	6	0	6	30	3	4	6	0	0	0	4	1	1	0	0	1	1
19	18	78.48562	14.67275	1	3	2	1	6	09:15	12	E0900002(	3	3218	6	30	3	4	3	4	0	0	1	1	1	0	0	1	1
20	19	78.59612	14.76983	1	3	2	1	6	21:16	12	E0900002(	4	302	6	30	0	-1	-1	0	0	0	4	1	1	0	0	1	1
21	20	78.61996	14.7222	1	2	1	1	7	03:00	12	E0900002(	3	4	6	30	3	4	6	0	0	0	4	1	1	0	0	1	1
22	21	78.57722	14.68994	1	3	1	1	2	21:45	12	E0900002(	6	0	6	30	0	-1	-1	0	0	0	4	1	1	0	0	1	2
23	22	78.60245	14.78535	1	3	2	1	2	17:05	12	E0900002(	4	415	6	30	3	4	5	0	0	0	4	1	1	0	0	1	2
24	23	78.64224	14.69112	1	3	2	1	2	21:30	12	E0900002(	3	3220	3	30	6	2	3	3220	0	0	4	1	4	0	0	1	1
25	24	78.63018	14.68432	1	3	2	1	3	17:25	12	E0900002(	5	0	6	30	6	4	6	0	0	0	4	1	1	0	0	1	2
26	25	78.6196	14.71209	1	3	2	1	7	07:34	12	E0900002(	3	4	3	30	3	2	4	319	0	5	1	1	2	0	0	1	1
27	26	78.51961	14.74997	1	3	1	1	4	16:35	12	E0900002(	5	0	6	30	3	4	6	0	0	1	4	2	2	0	0	1	1
28	27	78.54789	14.75609	1	3	2	1	1	20:00	12	E0900002(	4	450	6	30	0	-1	-1	0	0	0	4	1	1	0	0	1	1
29	28	78.48281	14.76918	1	3	2	2	7	13:15	12	E0900002(	3	3220	6	30	6	2	3	3218	0	5	1	2	2	0	0	1	3
30	29	78.48775	14.76714	1	3	2	1	3	18:20	12	E0900002(	3	4	3	30	6	2	5	0	0	5	4	2	2	0	0	1	1
31	30	78.62494	14.66271	1	3	2	1	4	07:25	12	E0900002(	4	319	6	30	3	4	6	0	0	1	4	1	1	0	0	1	1
32	31	78.63753	14.71344	1	2	2	2	3	17:30	12	E0900002(	4	412	2	30	2	4	6	0	0	0	4	2	2	0	0	1	1
33	32	78.50777	14.75667	1	3	2	1	5	12:30	1	E0900003:	3	4207	6	30	3	4	5	0	0	1	1	9	1	0	0	1	2
34	33	78.59393	14.77905	1	3	2	1	7	09:55	12	E0900002(	3	4	6	30	3	4	6	0	0	4	1	1	2	0	0	1	1
35	34	78.53039	14.71063	1	3	2	2	5	13:00	12	E0900002(	4	450	6	30	6	4	5	0	0	1	1	2	2	0	0	1	1
36	35	78.4772	14.69048	1	3	4	1	3	13:02	12	E0900002(	5	0	6	30	3	4	6	0	0	1	1	1	1	0	0	1	1
37	36	78.48284	14.75576	1	3	2	2	5	22:55	12	E0900002(	6	0	6	30	6	4	6	0	0	0	4	1	1	0	0	1	1

<u>2. Casualties Data</u> (`CasualtiesBig.csv`): Provides details about casualties involved in accidents.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Accident_	Vehicle_Re	Casualty_I	Casualty_(	Sex_of_Ca	Age_of_Ca	Age_Band	Casualty_S	Pedestriar	Pedestriar	Car_Passe	Bus_or_Cc	Pedestriar	Casualty_T	Casualty_H
2	1	1	1	3	1	37	7	2	1	1	0	0	-1	0	1
3	2	1	1	2	1	37	7	3	0	0	0	4	-1	11	1
4	3	2	1	1	1	62	9	3	0	0	0	0	-1	9	1
5	4	1	1	3	1	30	6	3	5	2	0	0	-1	0	1
6	5	1	1	1	1	49	8	3	0	0	0	0	-1	3	-1
7	6	2	1	1	2	30	6	3	0	0	0	0	-1	3	1
8	7	1	1	1	1	31	6	3	0	0	0	0	-1	3	-1
9	8	1	1	3	2	13	3	3	6	9	0	0	-1	0	1
10	9	1	2	3	2	13	3	3	6	9	0	0	-1	0	1
11	10	1	1	1	1	35	6	3	0	0	0	0	-1	9	1
12	11	2	2	1	2	48	8	3	0	0	0	0	-1	9	1
13	12	1	1	2	2	26	6	3	0	0	0	4	-1	11	1
14	13	1	2	2	2	9	2	3	0	0	0	4	-1	11	1
15	14	1	3	2	2	40	7	3	0	0	0	4	-1	11	1
16	15	1	4	2	2	38	7	3	0	0	0	4	-1	11	1
17	16	1	5	2	2	28	6	3	0	0	0	4	-1	11	-1
18	17	1	1	3	2	23	5	3	1	4	0	0	-1	0	1
19	18	2	1	1	2	20	4	3	0	0	0	0	-1	3	-1
20	19	1	1	3	2	75	10	3	1	1	0	0	-1	0	-1
21	20	1	1	1	1	34	6	3	0	0	0	0	-1	9	1
22	21	1	1	1	1	42	7	3	0	0	0	0	-1	5	1
23	22	1	2	3	2	31	6	3	5	3	0	0	-1	0	-1

<u>3. Vehicles Data</u> (`VehiclesBig.csv`): Contains information about vehicles involved in accidents.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Accident_I	Vehicle_Re	Vehicle_Ty	Towing_ar	Vehicle_M	Vehicle_Lc	Junction_L	Skidding_a	Hit_Object	Vehicle_Le	Hit_Object	1st_Point_	Was_Vehi	Journey_P	Sex_of_Dr	Age_of_Dr	Age_Band_	Engine_Ca	Propulsion	Age_of_Ve	Driver_IMI	Driver_Hor
2	1	1	9	0	18	0	0	0	0	0	0	1	1	15	2	74	10	-1	-1	-1	7	1
3	2	1	11	0	4	0	3	0	0	0	0	4	1	1	1	42	7	8268	2	3	-1	-1
4	3	1	11	0	17	0	0	0	4	0	0	4	1	1	1	35	6	8300	2	5	2	1
5	4	2	9	0	2	0	0	0	0	0	0	3	1	15	1	62	9	1762	1	6	1	1
6	5	1	9	0	18	0	0	0	0	0	0	1	1	15	2	49	8	1769	1	4	2	1
7	6	1	3	0	18	0	0	1	10	0	0	1	1	15	1	49	8	85	1	10	-1	-1
8	7	1	9	0	5	0	0	0	0	0	0	0	1	15	1	51	8	2976	1	1	4	1
9	8	2	3	0	18	0	0	0	0	0	0	0	1	15	2	30	6	124	1	2	1	1
10	9	1	3	0	18	0	1	0	4	0	0	1	1	15	1	31	6	-1	-1	-1	-1	-1
11	10	2	9	0	2	0	1	0	0	0	0	2	1	15	1	41	7	4266	1	4	6	1
12	11	1	9	0	18	0	0	0	0	1	0	1	1	15	1	68	10	5343	1	16	6	1
13	12	1	9	0	18	0	8	0	0	0	0	2	1	15	1	35	6	1998	1	13	3	1
14	13	2	9	0	9	0	8	0	0	0	0	1	1	15	2	48	8	-1	-1	-1	5	1
15	14	1	11	0	9	0	8	0	0	0	0	0	1	1	1	42	7	8268	2	2	2	1
16	15	2	90	0	18	0	8	0	0	0	0	0	1	15	3	-1	-1	-1	-1	-1	-1	-1
17	16	1	9	0	18	0	8	0	0	0	0	1	1	15	1	34	6	1988	1	6	4	1
18	17	1	9	0	9	0	8	0	0	0	0	1	1	15	2	19	4	1124	1	8	8	1
19	18	2	3	0	18	0	8	0	0	0	0	1	1	15	2	20	4	124	1	1	-1	-1
20	19	1	9	0	9	0	3	0	0	0	0	1	1	15	2	47	8	1360	1	2	3	1
21	20	1	9	0	18	0	8	0	7	7	1	1	1	15	1	34	6	698	1	2	7	1
22	21	2	9	0	7	0	8	0	0	0	0	0	1	15	1	33	6	2148	2	4	4	1
23	22	1	5	0	18	0	0	0	0	0	0	1	1	15	1	42	7	1171	1	2	8	1
24	23	1	11	0	18	0	1	0	0	0	0	4	1	1	1	57	9	7300	2	1	3	1
25	24	1	5	0	18	0	8	0	0	0	0	1	1	15	1	54	8	599	1	6	5	1
26	25	2	9	0	9	0	8	0	0	0	0	3	2	15	1	44	7	-1	-1	-1	8	1
27	26	1	9	0	9	0	8	0	0	0	0	3	1	15	1	27	6	1998	1	7	6	1

<u>Data Analysis</u>

1. Data Loading:

Loaded the three datasets into pandas dataframes.

2. Data Merging:

Merged the datasets based on the common key "Accident_Index."

3. Feature Selection:

Selected relevant features for analysis and modeling.

Machine Learning Model

1. Data Preprocessing:

Handled missing values and encoded categorical variables.

2. Model Training:

a. Feature Selection:

Selected relevant features (latitude, longitude, number of casualties) as input variables (X) for training the model.

b. Target Variable:

The target variable (y) was the accident severity, and the model was trained to predict this variable based on the selected features.

c. Training-Testing Split:

Split the dataset into training and testing sets to evaluate the model's performance. Typically, a common split ratio is 80% for training and 20% for testing.

d. Model Training:

Used the training set to train the Decision Tree classifier using the selected features and target variable.

3. Model Evaluation:

Evaluated the model's accuracy on a test set.

4. Prediction Output:

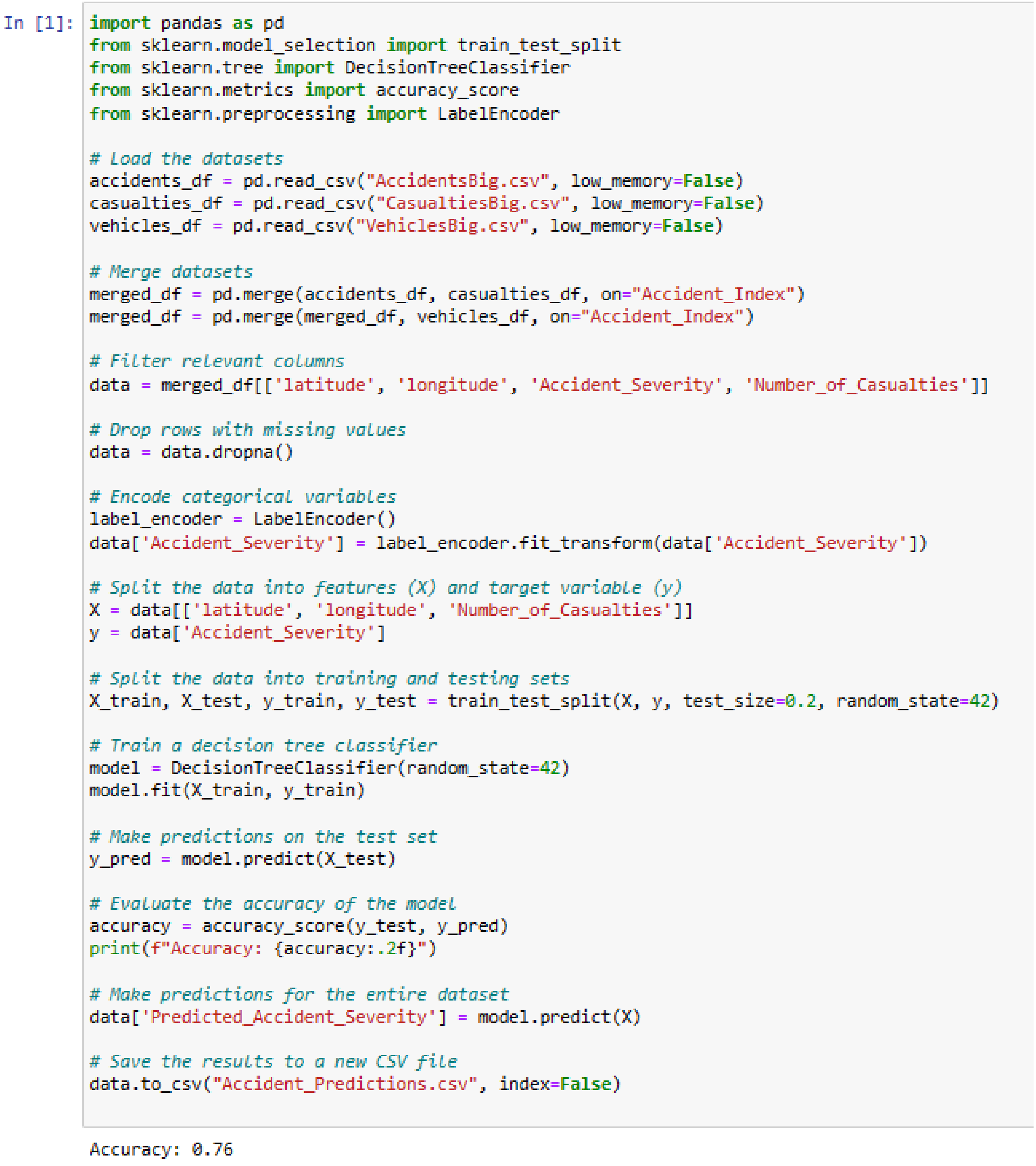
Added the predicted severity to the dataset.

5. Future Improvements:

Algorithm Exploration:

Consider exploring other machine learning algorithms (e.g., Random Forest, Gradient Boosting) to compare and improve prediction accuracy.

Code



In [1]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder

# Load the datasets
accidents_df = pd.read_csv("AccidentsBig.csv", low_memory=False)
casualties_df = pd.read_csv("CasualtiesBig.csv", low_memory=False)
vehicles_df = pd.read_csv("VehiclesBig.csv", low_memory=False)

# Merge datasets
merged_df = pd.merge(accidents_df, casualties_df, on="Accident_Index")
merged_df = pd.merge(merged_df, vehicles_df, on="Accident_Index")

# Filter relevant columns
data = merged_df[['latitude', 'longitude', 'Accident_Severity', 'Number_of_Casualties']]

# Drop rows with missing values
data = data.dropna()

# Encode categorical variables
label_encoder = LabelEncoder()
data['Accident_Severity'] = label_encoder.fit_transform(data['Accident_Severity'])

# Split the data into features (X) and target variable (y)
X = data[['latitude', 'longitude', 'Number_of_Casualties']]
y = data['Accident_Severity']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a decision tree classifier
model = DecisionTreeClassifier(random_state=42)
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")

# Make predictions for the entire dataset
data['Predicted_Accident_Severity'] = model.predict(X)

# Save the results to a new CSV file
data.to_csv("Accident_Predictions.csv", index=False)
```

```
Accuracy: 0.76
```

The accuracy of a machine learning model, such as the Decision Tree classifier, is a measure of how well the model performs on the test dataset. Specifically, accuracy is calculated as the ratio of correctly predicted instances to the total instances in the test set. It is often expressed as a percentage.

In our scenario, with an accuracy of **0.76 (or 76%)**, it means that the model correctly predicted the accident severity for approximately 76% of the instances in the test dataset.

Interpreting Accuracy:
High Accuracy (Close to 1):

A high accuracy indicates that the model is making correct predictions for a large portion of the test dataset.
Low Accuracy (Close to 0):

A low accuracy suggests that the model's predictions do not align well with the actual outcomes in the test set.

- **Factors Affecting Accuracy:**
  1. **Feature Relevance:**

     The choice of features used to train the model plays a crucial role. If latitude, longitude, and the number of casualties are relevant predictors, the model is more likely to perform well.
  2. **Dataset Characteristics:**

     The overall quality and characteristics of the dataset impact model performance. A well-prepared and representative dataset generally leads to better accuracy.
  3. **Model Complexity:**

     Decision Trees have hyperparameters that control their complexity. Adjusting these hyperparameters (e.g., tree depth) can impact accuracy.

Overly complex models may overfit the training data and not generalize well to new data.

4. **Data Split:**

   The way the dataset is split into training and testing sets can influence accuracy. Randomness in the split can cause variations in performance.

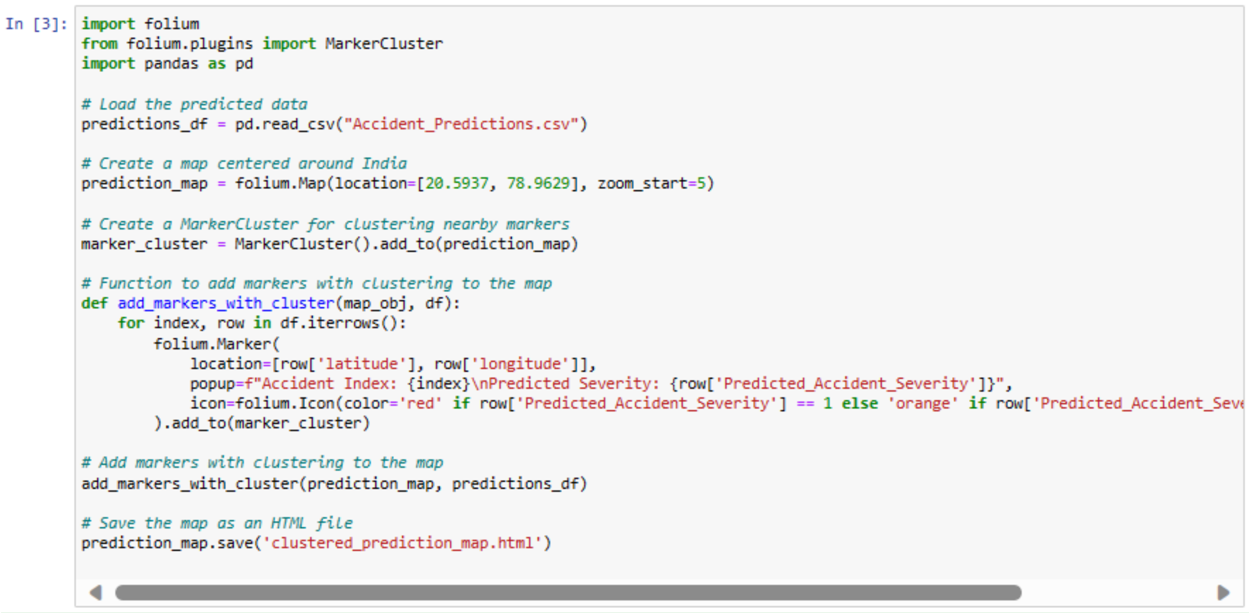
Map Visualization

1. Folium Integration:
   - Utilized Folium to create an interactive map.

2. Marker Clustering:
   - Implemented marker clustering for improved map performance.

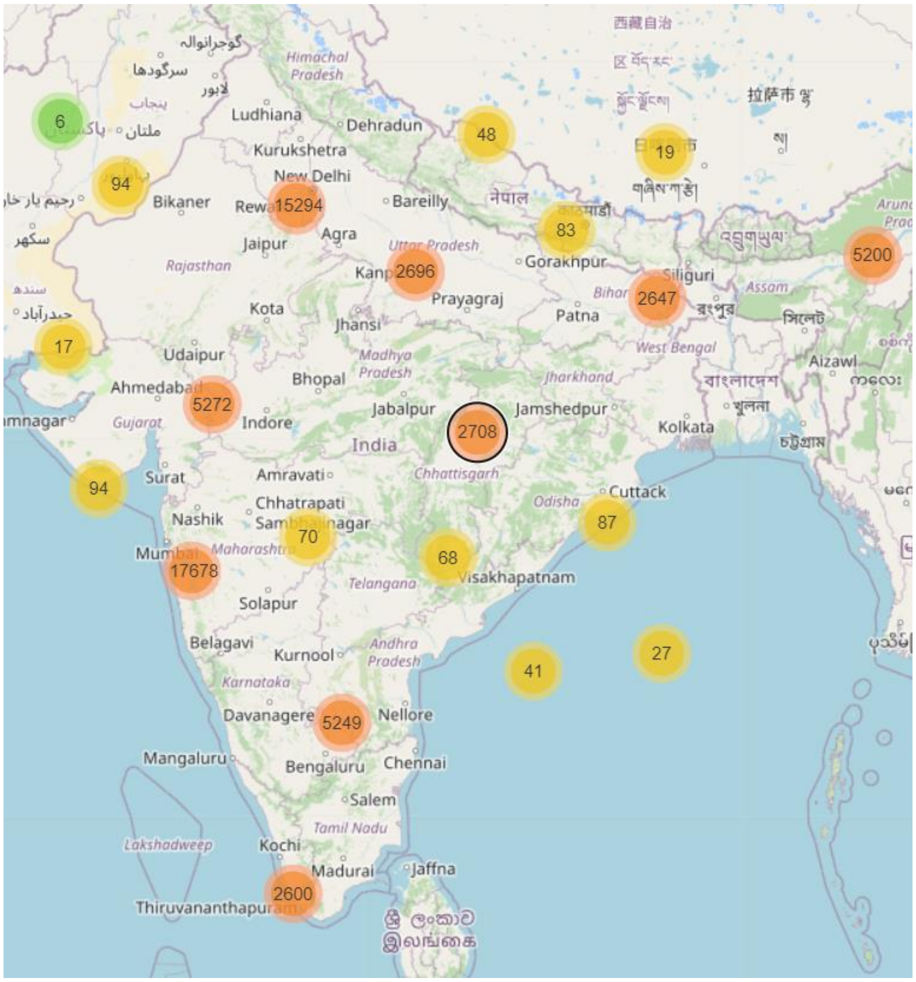
3. Map Output:
   - Saved the map as an HTML file (`clustered_prediction_map.html`).

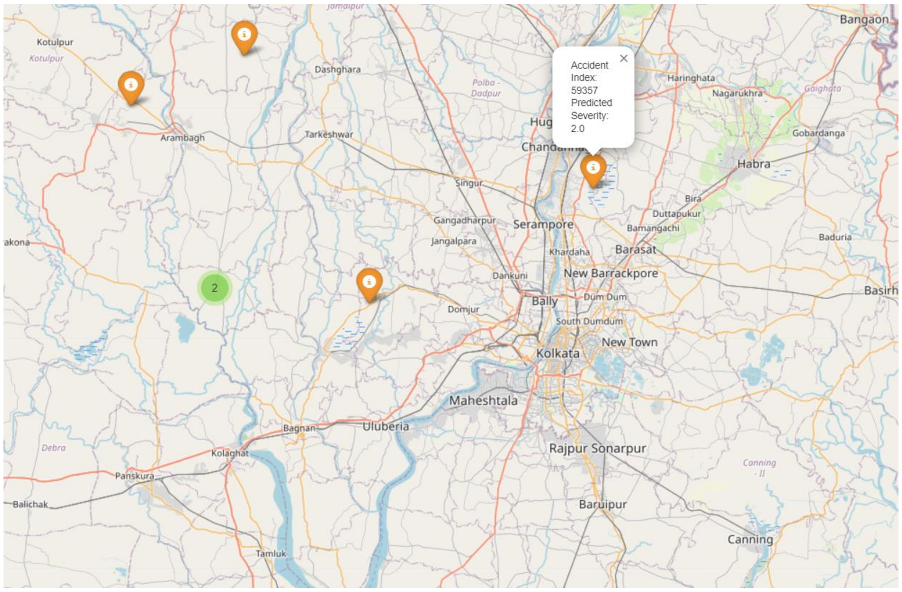
Code

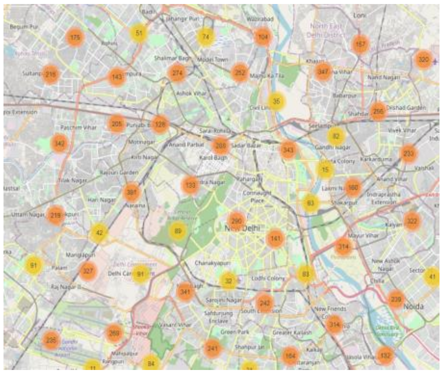


```
In [3]: import folium
        from folium.plugins import MarkerCluster
        import pandas as pd

        # Load the predicted data
        predictions_df = pd.read_csv("Accident_Predictions.csv")

        # Create a map centered around India
        prediction_map = folium.Map(location=[20.5937, 78.9629], zoom_start=5)

        # Create a MarkerCluster for clustering nearby markers
        marker_cluster = MarkerCluster().add_to(prediction_map)

        # Function to add markers with clustering to the map
        def add_markers_with_cluster(map_obj, df):
            for index, row in df.iterrows():
                folium.Marker(
                    location=[row['latitude'], row['longitude']],
                    popup=f"Accident Index: {index}\nPredicted Severity: {row['Predicted_Accident_Severity']}",
                    icon=folium.Icon(color='red' if row['Predicted_Accident_Severity'] == 1 else 'orange' if row['Predicted_Accident_Seve
                ).add_to(marker_cluster)

        # Add markers with clustering to the map
        add_markers_with_cluster(prediction_map, predictions_df)

        # Save the map as an HTML file
        prediction_map.save('clustered_prediction_map.html')
```

Output Map




Accident Index: 59357 Predicted Severity: 2.0



Project Utilization Across Organizations:

1. Police Departments:

a. Accident Hotspots:

Utilization: Identify areas with a high probability of accidents based on historical data.

Benefit: Enable law enforcement to allocate resources strategically, conduct targeted patrols, and enhance accident prevention efforts.

b. Severity Prediction:

Utilization: Predict the severity of accidents in real-time.

Benefit: Enable quick response and resource mobilization, especially for high-severity incidents, and improve emergency services.

c. Trend Analysis:

Utilization: Analyze historical data to identify trends and patterns.

Benefit: Support policy decisions, allocate resources efficiently, and implement targeted safety measures.

2. National Highway Authority of India (NHAI):

a. Infrastructure Planning:

Utilization: Identify accident-prone zones to inform infrastructure planning.

Benefit: Support NHAI in designing road layouts, installing safety features, and implementing measures to reduce accidents.

b. Real-time Monitoring:

Utilization: Utilize real-time predictions for monitoring accident situations.

Benefit: Facilitate prompt responses, rerouting traffic, and managing road conditions during emergencies.

c. Public Awareness:

Utilization: Leverage data insights to create public awareness campaigns.
Benefit: Educate drivers and pedestrians about accident-prone areas, safe driving practices, and adherence to traffic rules.

3. Highway Authorities:

a. Maintenance Planning:

Utilization: Identify areas requiring maintenance based on accident data.
Benefit: Prioritize maintenance activities to enhance road conditions and reduce accident risks.

b. Traffic Management:

Utilization: Use real-time predictions for efficient traffic management.
Benefit: Improve traffic flow, implement dynamic speed limits, and reduce congestion in accident-prone areas.

c. Emergency Response Coordination:

Utilization: Coordinate emergency response activities using severity predictions.
Benefit: Streamline response efforts, allocate resources effectively, and minimize response time.

5. Self-Driving Cars and Modern Vehicles:

a. Accident-Aware Navigation:

Utilization: Integrate the project's live API into self-driving cars and modern vehicles' navigation systems. Benefit: Provide real-time information on accident-prone regions to the vehicle's navigation system, enabling route optimization and avoidance of high-risk areas.

b. Dynamic Speed Limiting:

Utilization: Utilize severity predictions to dynamically adjust vehicle speed limits.

Benefit: Enhance safety by automatically limiting the speed of vehicles in areas with a higher likelihood of accidents, reducing the risk of collisions and improving overall road safety.

c. Collision Prevention Systems:
Utilization: Integrate severity predictions into collision prevention systems.
Benefit: Enable vehicles to proactively adjust their behavior, such as maintaining a safer following distance or activating advanced safety features, in response to the predicted severity of accidents in the vicinity.

d. Data-Driven Decision Making:
Utilization: Provide vehicles with access to historical accident data and predictive analytics.
Benefit: Equip vehicles with the ability to make data-driven decisions, enhancing the overall safety and efficiency of autonomous and connected vehicle systems.

Conclusion

The project successfully predicts accident zones based on historical data and visualizes the predictions on an interactive map. The machine learning model can be further refined for enhanced accuracy, and additional features can be explored for a comprehensive analysis.