



Project Title	<b>Regulatory Affairs of Road Accident Data 2020 India</b>
Tools	Python, ML, SQL, Excel
Technologies	Data Analyst & Data scientist
Project Difficulties level	intermediate

Dataset : Dataset is available in the given link. You can download it at your convenience.

[Click here to download data set](#)

## About Dataset

The data contains road accident information for 50 cities of India in the year 2020.

source : <https://data.gov.in/catalog/road-accidents-india-2020>

Columns :

1. Million plus cities : Contains the name of the cities
2. Cause Category: 5 primary category for classification of accidents ( Traffic Control, Junction, Road Features, Impacting Vehicle/Object, Weather Conditions)
3. Cause Subcategory: Further classifying the exact cause for the accident
4. Outcome of Incident: Indicating , injuries , deaths and accidents
5. Count: Count in Millions for each incident.

#### NOTE :

1. this project is only for your guidance, not exactly the same you have to create. Here I am trying to show the way or idea of what steps you can follow and how your projects look. Some projects are very advanced (because it will be made with the help of flask, nlp, advance ai, advance DL and some advanced things ) which you can not understand .
2. You can make or analyze your project with yourself, with your idea, make it more creative from where we can get some information and understand about our business. make sure what overall things you have created all things you understand very well.

## Example

what steps you should have to follow

For your project on Road Accident Data 2020 in India using the columns **Million Plus Cities**, **Cause category**, **Cause Subcategory**, **Outcome of Incident**, and **Count**, here's a structured approach with code examples:

### Project Title:

### **Analysis of Road Accident Causes and Outcomes in Million-Plus Cities of India (2020)**

#### 1. Objective

The goal is to analyze the causes of road accidents in million-plus cities in India, identify patterns in causes and outcomes, and visualize the distribution of accidents based on different categories.

#### 2. Data Preparation

#### Python Code:

```
import pandas as pd

# Load the dataset
df = pd.read_csv('road_accident_data_2020.csv')
```

```
# Inspect the first few rows of the dataset
print(df.head())
```

```
# Check for missing values
print(df.isnull().sum())
```

### **Expected Output:**

- The first few rows of the dataset.
- A summary showing any missing values in each column.

## **3. Data Cleaning**

### **Python Code:**

```
# Drop rows with missing values if any
df_cleaned = df.dropna()
```

```
# Verify the cleaning process
print(df_cleaned.isnull().sum())
```

### **Expected Output:**

- A summary confirming that there are no missing values after cleaning.

## **4. Exploratory Data Analysis (EDA)**

### **A. Distribution of Accidents Across Cities**

#### **Python Code:**

```
import matplotlib.pyplot as plt
import seaborn as sns

# Plot the distribution of accidents by city
plt.figure(figsize=(12,6))
sns.countplot(y='Million Plus Cities', data=df_cleaned,
order=df_cleaned['Million Plus Cities'].value_counts().index)
plt.title('Distribution of Road Accidents in Million-Plus
Cities')
```

```
plt.xlabel('Number of Accidents')
plt.ylabel('Cities')
plt.show()
```

### Expected Output:

- A bar plot showing the number of accidents in each million-plus city.

### B. Analysis of Accident Causes

#### Python Code:

```
# Plot the distribution of accidents by cause category
plt.figure(figsize=(10,6))
sns.countplot(y='Cause category', data=df_cleaned,
order=df_cleaned['Cause category'].value_counts().index)
plt.title('Distribution of Accident Causes')
plt.xlabel('Number of Accidents')
plt.ylabel('Cause Category')
plt.show()

# Detailed analysis by cause subcategory
plt.figure(figsize=(10,8))
sns.countplot(y='Cause Subcategory', data=df_cleaned,
order=df_cleaned['Cause Subcategory'].value_counts().index)
plt.title('Detailed Analysis of Accident Causes by
Subcategory')
plt.xlabel('Number of Accidents')
plt.ylabel('Cause Subcategory')
plt.show()
```

### Expected Output:

- A bar plot showing the distribution of accident causes by category.
- A more detailed bar plot showing the distribution of accident causes by subcategory.

### C. Outcomes of Incidents

#### Python Code:

```
# Plot the outcomes of incidents
plt.figure(figsize=(8,5))
sns.countplot(x='Outcome of Incident', data=df_cleaned,
order=df_cleaned['Outcome of Incident'].value_counts().index)
plt.title('Outcome of Road Accidents')
plt.xlabel('Outcome')
plt.ylabel('Number of Incidents')
plt.show()
```

### **Expected Output:**

- A bar plot showing the distribution of outcomes (e.g., fatal, non-fatal) of road accidents.

## **5. Analyzing Accident Causes vs Outcomes**

### **Python Code:**

```
# Grouping by cause category and outcome
outcome_vs_cause = df_cleaned.groupby(['Cause category',
'Outcome of Incident'])['Count'].sum().unstack()

# Plot the result
outcome_vs_cause.plot(kind='bar', stacked=True, figsize=(12,8))
plt.title('Accident Causes vs Outcomes')
plt.xlabel('Cause Category')
plt.ylabel('Number of Incidents')
plt.show()
```

### **Expected Output:**

- A stacked bar plot showing the relationship between accident causes and their outcomes.

## **6. Conclusion**

Summarize key findings from the analysis:

- Identify the cities with the highest number of accidents.

- Highlight the most common causes of accidents.
- Discuss the relationship between causes and outcomes of road accidents.

This project will help in understanding the patterns in road accidents, which can inform policy-making for safer roads in India's million-plus cities.

4o  
ChatGPT can

## Sample code

```
import pandas as pd
import numpy as np
```

In [2]:

```
df=pd.read_csv("/kaggle/input/road-accident-data-2020-india/df.csv")
df
```

Out[2]:

	Million Plus Cities	Cause category	Cause Subcategory	Outcome of Incident	Count
0	Agra	Traffic Control	Flashing Signal/Blinker	Grievously Injured	0.0
1	Agra	Traffic Control	Flashing Signal/Blinker	Minor Injury	0.0

2	Agra	Traffic Control	Flashing Signal/Blinker	Persons Killed	0.0
3	Agra	Traffic Control	Flashing Signal/Blinker	Total Injured	0.0
4	Agra	Traffic Control	Flashing Signal/Blinker	Total number of Accidents	0.0
...	...	...	...	...	...
9545	Vizaq	Weather	Sunny/Clear	Greviously Injured	561.0
9546	Vizaq	Weather	Sunny/Clear	Minor Injury	252.0
9547	Vizaq	Weather	Sunny/Clear	Persons Killed	176.0
9548	Vizaq	Weather	Sunny/Clear	Total number of Accidents	1207.0
9549	Vizaq	Weather	Sunny/Clear	Total Injured	813.0

9550 rows × 5 columns

df.shape

In [3]:

Out[3]:

(9550, 5)

In [4]:

df.size

Out[4]:

47750

In [5]:

df.head()

Out[5]:

	Million Plus Cities	Cause category	Cause Subcategory	Outcome of Incident	Count
0	Agra	Traffic Control	Flashing Signal/Blinker	Greviously Injured	0.0
1	Agra	Traffic Control	Flashing Signal/Blinker	Minor Injury	0.0
2	Agra	Traffic Control	Flashing Signal/Blinker	Persons Killed	0.0



3	Agra	Traffic Control	Flashing Signal/Blinker	Total Injured	0.0
4	Agra	Traffic Control	Flashing Signal/Blinker	Total number of Accidents	0.0

In [6]:

```
df.tail()
```



Out[6]:

	Million Plus Cities	Cause category	Cause Subcategory	Outcome of Incident	Count
9545	Vizaq	Weather	Sunny/Clear	Previously Injured	561.0
9546	Vizaq	Weather	Sunny/Clear	Minor Injury	252.0
9547	Vizaq	Weather	Sunny/Clear	Persons Killed	176.0
9548	Vizaq	Weather	Sunny/Clear	Total number of Accidents	1207.0
9549	Vizaq	Weather	Sunny/Clear	Total Injured	813.0

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 9550 entries, 0 to 9549
```

```
Data columns (total 5 columns):
```

#	Column	Non-Null Count	Dtype
0	Million Plus Cities	9550 non-null	object
1	Cause category	9550 non-null	object
2	Cause Subcategory	9550 non-null	object
3	Outcome of Incident	9550 non-null	object
4	Count	9547 non-null	float64

```
dtypes: float64(1), object(4)
```

```
memory usage: 373.2+ KB
```

In [8]:

```
df.columns
```

Out[8]:

```
Index(['Million Plus Cities', 'Cause category', 'Cause Subcategory',  
      'Outcome of Incident', 'Count'],  
      dtype='object')
```

In [9]:

```
df.isnull().sum()
```

Out[9]:

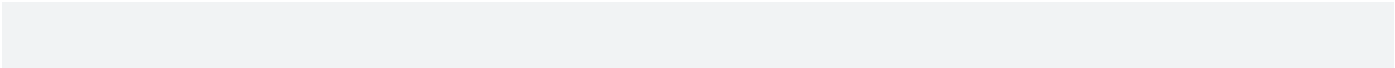
```
Million Plus Cities    0  
Cause category         0
```

Cause Subcategory        0  
Outcome of Incident       0  
Count                      3

dtype: int64

In [10]:

df.fillna(0)



Out[10]:

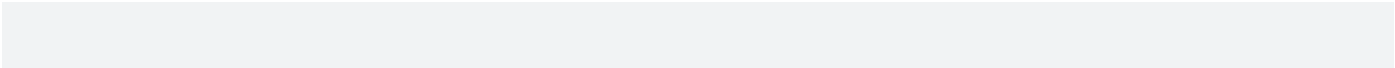
	Million Plus Cities	Cause category	Cause Subcategory	Outcome of Incident	Count
0	Agra	Traffic Control	Flashing Signal/Blinker	Greviously Injured	0.0
1	Agra	Traffic Control	Flashing Signal/Blinker	Minor Injury	0.0
2	Agra	Traffic Control	Flashing Signal/Blinker	Persons Killed	0.0
3	Agra	Traffic Control	Flashing Signal/Blinker	Total Injured	0.0
4	Agra	Traffic Control	Flashing Signal/Blinker	Total number of Accidents	0.0
...	...	...	...	...	...

9545	Vizaq	Weather	Sunny/Clear	Greviously Injured	561.0
9546	Vizaq	Weather	Sunny/Clear	Minor Injury	252.0
9547	Vizaq	Weather	Sunny/Clear	Persons Killed	176.0
9548	Vizaq	Weather	Sunny/Clear	Total number of Accidents	1207.0
9549	Vizaq	Weather	Sunny/Clear	Total Injured	813.0

9550 rows × 5 columns

In [11]:

```
df["Million Plus Cities"].value_counts()
```



Out[11]:

Million Plus Cities

Agra	191
Patna	191
Kollam	191
Kota	191
Lucknow	191
Ludhiana	191
Madurai	191
Mallapuram	191

Meerut	191
Mumbai	191
Nagpur	191
Nashik	191
Pune	191
Ahmedabad	191
Raipur	191
Rajkot	191
Srinagar	191
Surat	191
Thiruvanthapuram	191
Thrissur	191
Tiruchirapalli	191
Vadodra	191
Varanasi	191
Vijaywada city	191
Kolkata	191
Kochi	191
Khozikode	191
Kanpur	191
Allahabad(Prayagraj)	191
Amritsar	191
Asansol Durgapur	191
Aurangabad	191
Bengaluru	191
Bhopal	191
Chandigarh	191
Chennai	191
Coimbatore	191
Delhi	191
Dhanbad	191
Faridabad	191
Ghaziabad	191
Gwalior	191

Hyderabad	191
Indore	191
Jabalpur	191
Jaipur	191
Jamshedpur	191
Jodhpur	191
Kannur	191
Vizag	191

Name: count, dtype: int64

In [12]:

```
df["Cause category"].value_counts()
```

Out[12]:

Cause category	
Road Features	2000
Impacting Vehicle/Object	1800
Traffic Control	1500
Junction	1500
Traffic Violation	1500
Weather	1250

Name: count, dtype: int64

In [13]:

```
df["Cause Subcategory"].value_counts()
```

Out[13]:

Cause Subcategory	
Others	1450
Flashing Signal/Blinker	250

Over	250
Rainy	250
Hail/Sleet	250
Foggy and Misty	250
Straight Road	250
Steep Grade	250
Pot Holes	250
Ongoing Road Works/Under Construction	250
Curved Road	250
Bridge	250
Use of Mobile Phone	250
Culvert	250
Jumping Red Light	250
Four arm Junction	250
Police Controlled	250
Stop Sign	250
Traffic Light Signal	250
Drunken Driving/ Consumption of alcohol and drug	250
Uncontrolled	250
Sunny/Clear	250
Round about Junction	250
Staggered Junction	250
T	250
Y	250
Driving on Wrong side	250
Auto Rickshaws	200
Buses	200
Cars, Taxis, Vans and LMV	200
Other Non	200
Pedestrian	200
Trucks/Lorries	200
Two Wheelers	200
Bicycles	200

Name: count, dtype: int64

In [14]:

```
df["Outcome of Incident"].value_counts()
```

Out[14]:

```
Outcome of Incident
Greviously Injured      2000
Minor Injury            2000
Persons Killed          2000
Total number of Accidents 2000
Total Injured           1550
```

Name: count, dtype: int64

In [15]:

```
df=pd.read_csv("/kaggle/input/road-accident-data-2020-india/df.csv",index_col="Milli
on Plus Cities")
df
```

Out[15]:

	Cause category	Cause Subcategory	Outcome of Incident	Count
Million Plus Cities				
Agra	Traffic Control	Flashing Signal/Blinker	Greviously Injured	0.0

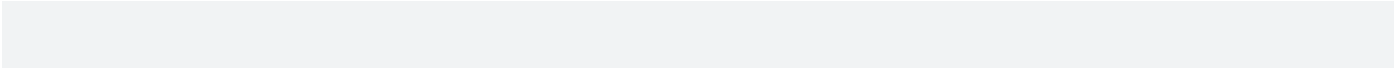


Agra	Traffic Control	Flashing Signal/Blinker	Minor Injury	0.0
Agra	Traffic Control	Flashing Signal/Blinker	Persons Killed	0.0
Agra	Traffic Control	Flashing Signal/Blinker	Total Injured	0.0
Agra	Traffic Control	Flashing Signal/Blinker	Total number of Accidents	0.0
...	...	...	...	...
Vizaq	Weather	Sunny/Clear	Greviously Injured	561.0
Vizaq	Weather	Sunny/Clear	Minor Injury	252.0
Vizaq	Weather	Sunny/Clear	Persons Killed	176.0
Vizaq	Weather	Sunny/Clear	Total number of Accidents	1207.0
Vizaq	Weather	Sunny/Clear	Total Injured	813.0

9550 rows × 4 columns

In [16]:

```
df.sort_index(ascending=False)
```



Out[16]:

	Cause category	Cause Subcategory	Outcome of Incident	Count
Million Plus Cities				
Vizaq	Weather	Sunny/Clear	Total Injured	813.0
Vizaq	Junction	Y	Greviously Injured	25.0
Vizaq	Traffic Violation	Over	Minor Injury	277.0
Vizaq	Traffic Violation	Over	Greviously Injured	590.0
Vizaq	Traffic Violation	Others	Total Injured	304.0
...	...	...	...	...

Agra	Traffic Violation	Use of Mobile Phone	Previously Injured	8.0
Agra	Traffic Violation	Use of Mobile Phone	Minor Injury	3.0
Agra	Traffic Violation	Use of Mobile Phone	Total number of Accidents	16.0
Agra	Traffic Violation	Use of Mobile Phone	Persons Killed	9.0
Agra	Traffic Control	Flashing Signal/Blinker	Previously Injured	0.0

9550 rows × 4 columns

```
In [17]:  
df  
  
[REDACTED]
```

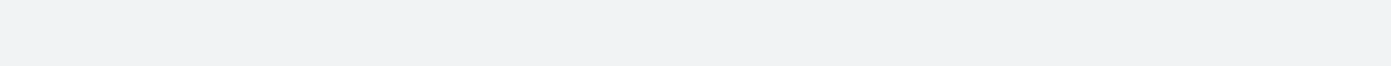
	Cause category	Cause Subcategory	Outcome of Incident	Count
Million Plus Cities				
Agra	Traffic Control	Flashing Signal/Blinker	Previously Injured	0.0

Agra	Traffic Control	Flashing Signal/Blinker	Minor Injury	0.0
Agra	Traffic Control	Flashing Signal/Blinker	Persons Killed	0.0
Agra	Traffic Control	Flashing Signal/Blinker	Total Injured	0.0
Agra	Traffic Control	Flashing Signal/Blinker	Total number of Accidents	0.0
...	...	...	...	...
Vizaq	Weather	Sunny/Clear	Greviously Injured	561.0
Vizaq	Weather	Sunny/Clear	Minor Injury	252.0
Vizaq	Weather	Sunny/Clear	Persons Killed	176.0
Vizaq	Weather	Sunny/Clear	Total number of Accidents	1207.0
Vizaq	Weather	Sunny/Clear	Total Injured	813.0

9550 rows × 4 columns

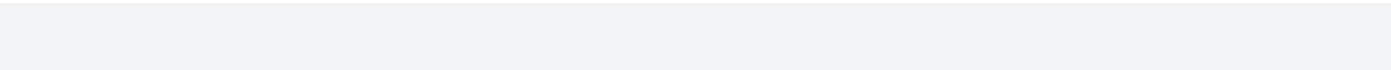
In [18]:

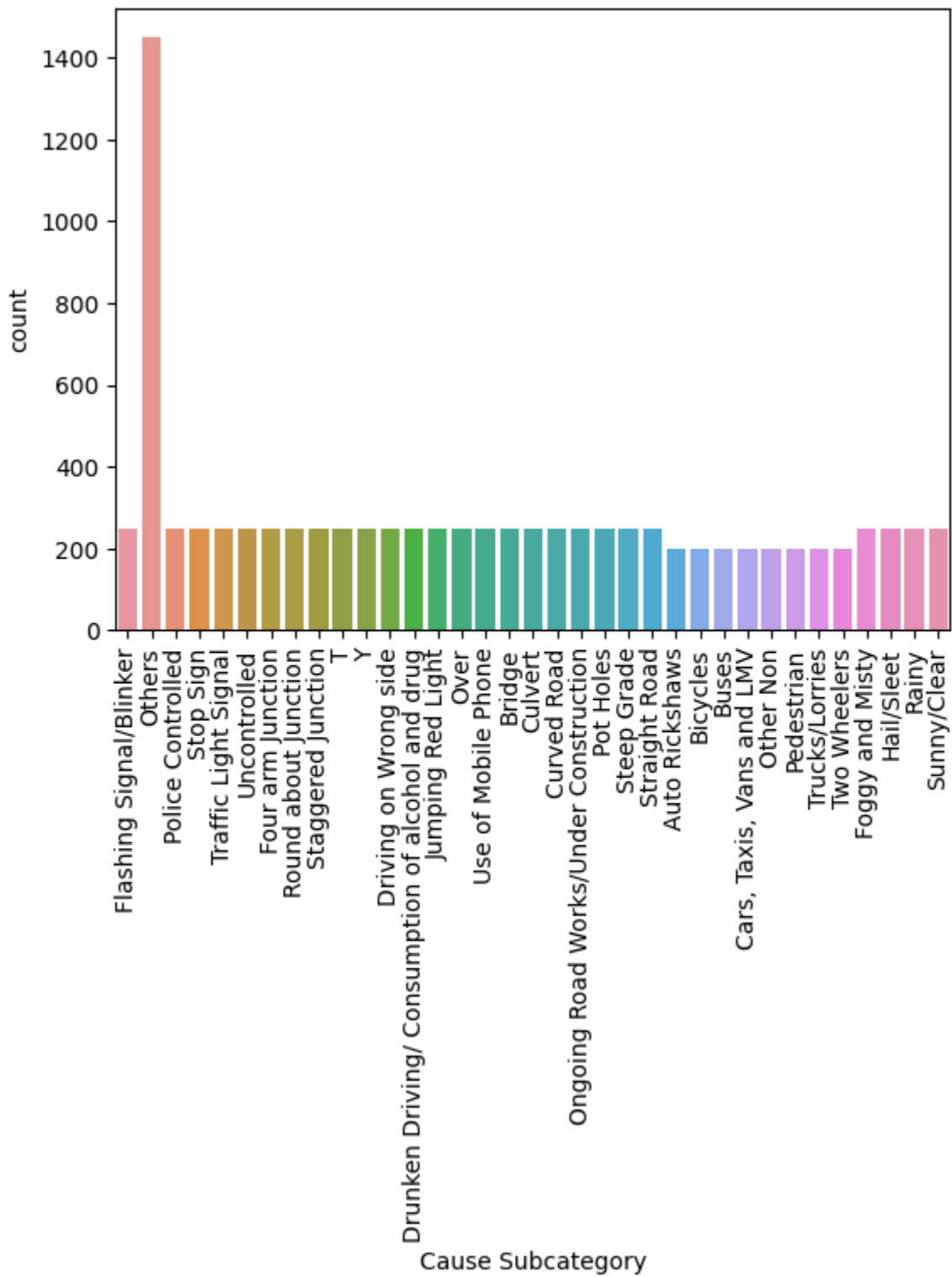
```
import matplotlib.pyplot as plt
import seaborn as sns
```



In [19]:

```
sns.countplot(data=df, x="Cause Subcategory")
plt.xticks(rotation=90)
plt.show()
```





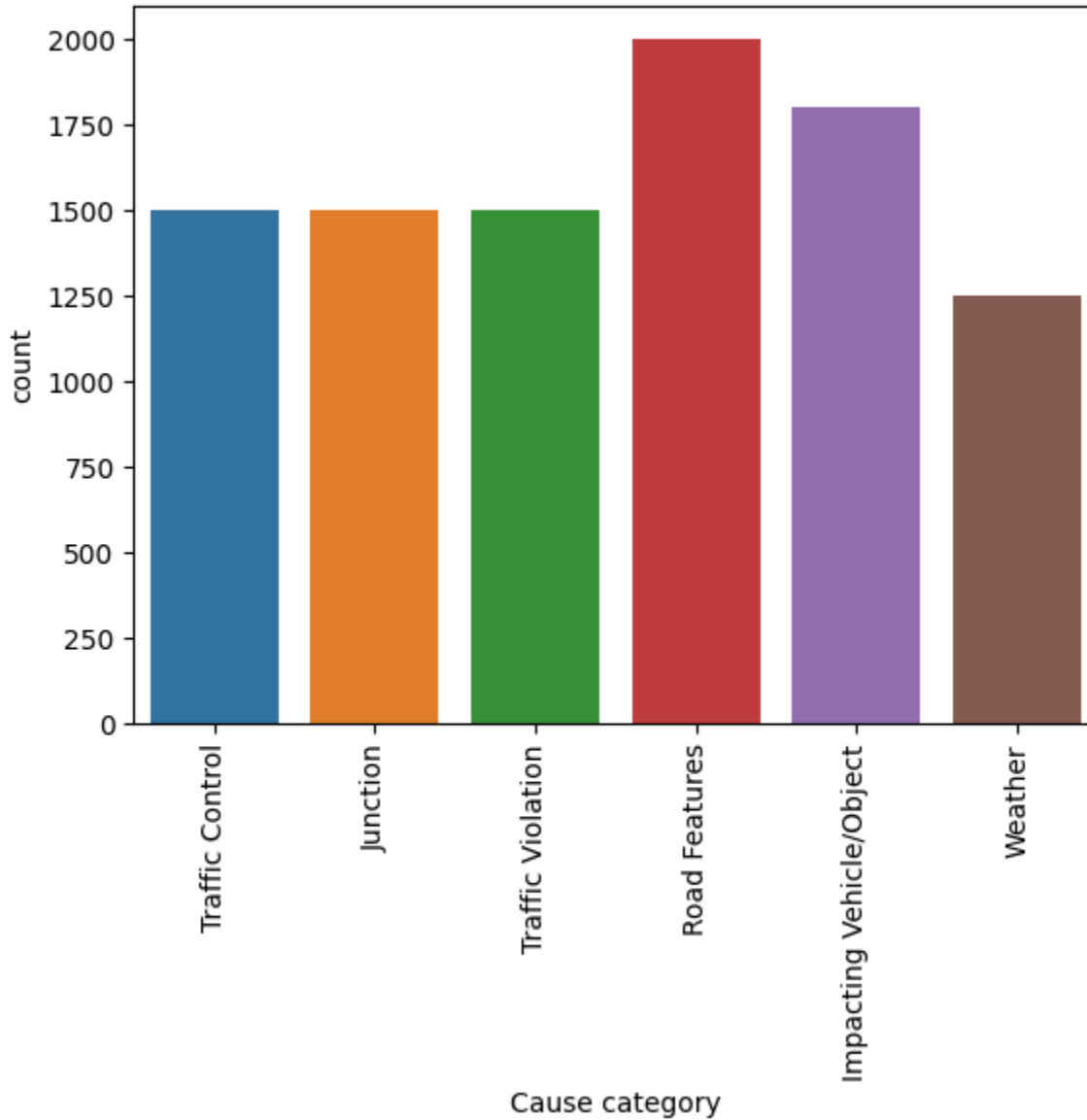
In [20]:

linkcode

```
sns.countplot(data=df, x="Cause category")
```

```
plt.xticks(rotation=90)
```

```
plt.show()
```



1 [Reference link](#)

2 [Reference link](#) for ML project