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

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, MinMaxScaler

# Load dataset
df = pd.read_csv("/content/AccidentsBig.csv")
```

```
<ipython-input-21-20dc5aed320e>:5: DtypeWarning: Columns (8,10,28,29) have mixed types. Specify dtype option on import or set low_me
  df = pd.read_csv("/content/AccidentsBig.csv")
```

```
df.head(5)
```

	Accident_Index	longitude	latitude	Police_Force	Accident_Severity	Number_of_Vehicles	Number_of_Casualties	Day_of_Week	Time
0	1.0	78.610393	14.724026	1.0	2.0	1.0	1.0	3.0	105
1	2.0	78.534042	14.762353	1.0	3.0	1.0	1.0	4.0	104
2	3.0	78.470877	14.745606	1.0	3.0	2.0	1.0	5.0	1
3	4.0	78.557994	14.667128	1.0	3.0	1.0	1.0	6.0	62
4	5.0	78.576431	14.703443	1.0	3.0	1.0	1.0	2.0	126

5 rows × 30 columns

```
df.describe()
```

	Accident_Index	longitude	latitude	Police_Force	Accident_Severity	Number_of_Vehicles	Number_of_Casualties	Day_of
count	59999.000000	59999.000000	59999.000000	59999.000000	59999.000000	59999.000000	59999.000000	59999.0
mean	0.500008	0.360205	0.595347	0.070308	0.927757	0.048943	0.014581	0.5
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
25%	0.250004	0.193549	0.436448	0.000000	1.000000	0.000000	0.000000	0.3
50%	0.500008	0.318662	0.624578	0.063830	1.000000	0.058824	0.000000	0.5
75%	0.750013	0.377905	0.820878	0.106383	1.000000	0.058824	0.000000	0.8
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0
std	0.288687	0.235938	0.254699	0.097217	0.190490	0.040753	0.034641	0.3

8 rows × 30 columns

```
# Drop duplicate rows
df.drop_duplicates(inplace=True)

# Handle missing values (drop columns with excessive missing data, impute others)
df.dropna(axis=1, thresh=int(0.7 * len(df)), inplace=True)  # Drop columns with >30% missing
df.fillna(method='ffill', inplace=True)  # Forward fill missing values
```

```
<ipython-input-22-6ac181687704>:6: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. U
  df.fillna(method='ffill', inplace=True)  # Forward fill missing values
```

```
df.drop_duplicates()
```

	Accident_Index	longitude	latitude	Police_Force	Accident_Severity	Number_of_Vehicles	Number_of_Casualties	Day_of_Week
0	0.000000	0.380511	0.263894	0.000000	0.5	0.000000	0.000000	0.333333
1	0.000017	0.377484	0.265620	0.000000	1.0	0.000000	0.000000	0.500000
2	0.000033	0.374979	0.264866	0.000000	1.0	0.058824	0.000000	0.666667
3	0.000050	0.378433	0.261331	0.000000	1.0	0.000000	0.000000	0.833333
4	0.000067	0.379165	0.262967	0.000000	1.0	0.000000	0.000000	0.166667
...	...	...	...	...	...	...	...	...
59993	0.999933	0.044796	0.708051	0.234043	1.0	0.058824	0.045455	0.500000
59994	0.999950	0.734237	0.762856	0.234043	1.0	0.058824	0.000000	0.500000
59995	0.999967	0.060730	0.756555	0.234043	1.0	0.058824	0.000000	0.500000
59996	0.999983	0.317852	0.740594	0.234043	1.0	0.117647	0.045455	0.500000
59997	1.000000	0.003163	0.579671	0.234043	1.0	0.058824	0.045455	0.500000

59998 rows × 30 columns

```
for col in df.select_dtypes(include=['number']).columns:
    mean_value = df[df[col] != 0][col].mean()  # Compute mean excluding zeros
    df[col] = df[col].replace(0, mean_value)
```

```
from scipy import stats
z_threshold = 3

# Compute Z-scores for all numeric columns
z_scores = np.abs(stats.zscore(df.select_dtypes(include=['number'])))

# Filter out rows with Z-score above the threshold
df_cleaned = df[(z_scores < z_threshold).all(axis=1)]

# Save cleaned dataset
df_cleaned.to_csv("cleaned_data.csv", index=False)

print("Outliers removed successfully using Z-score method!")
```

```
Outliers removed successfully using Z-score method!
```

```
from google.colab import files

# Save and download the cleaned dataset
df_cleaned.to_csv("cleaned_data.csv", index=False)
files.download("cleaned_data.csv")
```

Start coding or generate with AI.

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.