

TRAFFIC ACCIDENT DATA PIPELINE

From Raw Data to Actionable Insights

DATA PIPELINE WORKFLOW

● Data Ingestion

Validates and loads raw data into a structured format

● Data Cleaning

Removes duplicates, standardizes column names, and normalizes values

● Data Transformation

Enhances data with calculated metrics

● Data Aggregation

Summarizes data for trends and comparative analysis

● Exploratory Data Analysis (EDA)

Visualizes key patterns and trends

● Data Storage

Saves processed results for future use

● Automation

Integrates all steps into an automated workflow

DATA CLEANING PROCESS

- Duplicates were removed to maintain data integrity.
- Ensures consistent naming conventions.
- Groups similar entries accurately.

```
def clean_data(data): 1 usage • Reni Korueva
    # Remove duplicates
    data = data.drop_duplicates()

    # Ensure all column names are trimmed and consistent
    data.columns = data.columns.str.strip()

    # Standardize string values in categorical columns
    if 'Accident Type' in data.columns:
        data['Accident Type'] = data['Accident Type'].str.strip()
    if 'Road Safety Measures' in data.columns:
        data['Road Safety Measures'] = data['Road Safety Measures'].str.strip()

    return data
```



ADDING NEW INSIGHTS THROUGH TRANSFORMATION

- The Severity Score highlights accident seriousness.
- The Fatality Rate represents the percentage of fatalities per 100 accidents.
- The Injury Rate represents the percentage of injuries per 100 accidents.
- Impact: Enhanced dataset with deeper insights for decision-making.

```
def transform_data(data): 1 usage • Reni Korueva
# Severity score: weighted sum of fatalities and injuries
data['Severity Score'] = data['Fatalities'] * 3 + data['Injuries']

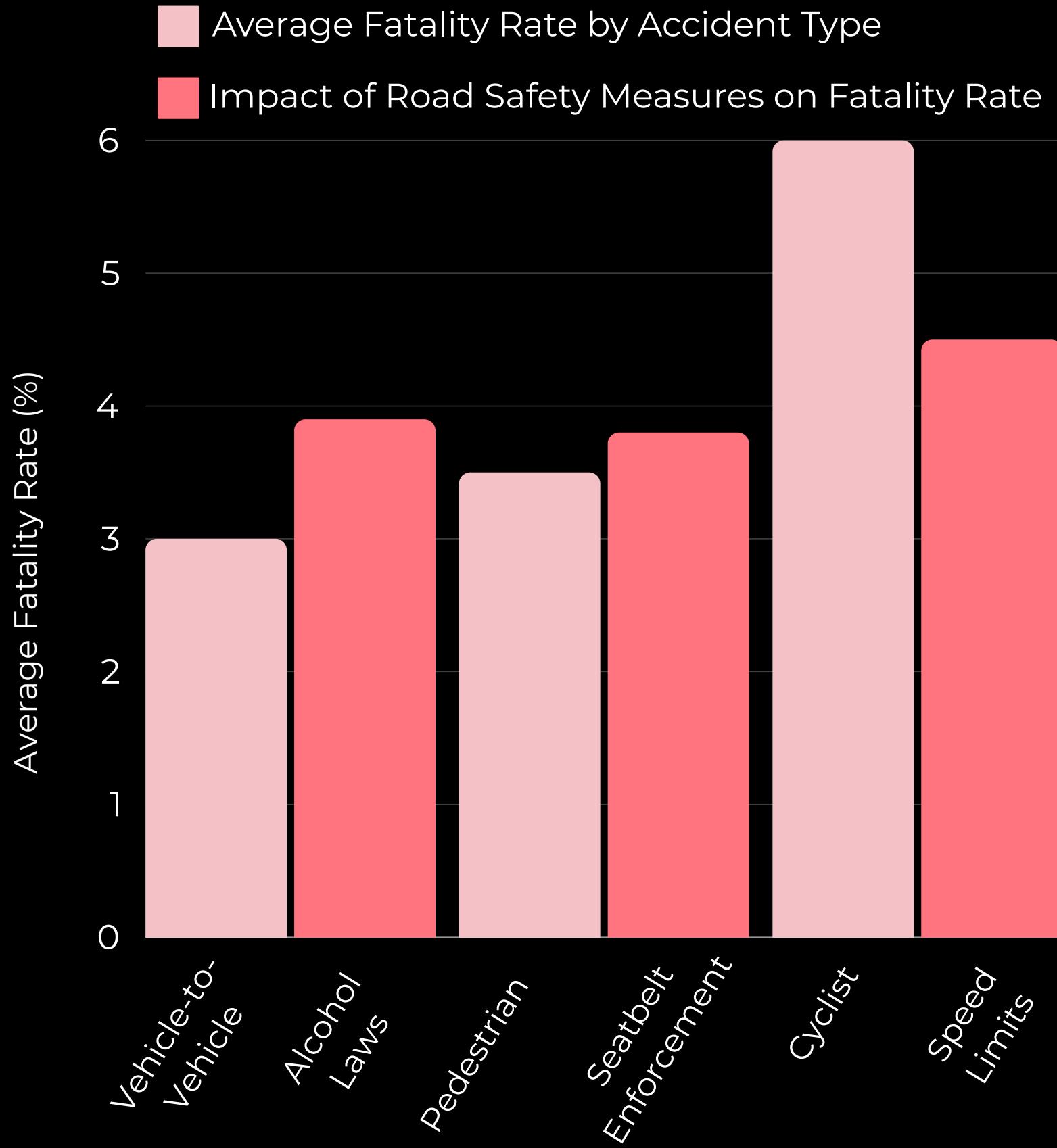
# Add rate features for fatality and injury
data['Fatality Rate'] = (data['Fatalities'] / data['Accidents Reported']).fillna(0) * 100
data['Injury Rate'] = (data['Injuries'] / data['Accidents Reported']).fillna(0) * 100
return data
```

EXTRACTING KEY TRENDS

- Grouped by Country and Year.
- Metrics computed:
 - Total and average accidents.
 - Total and mean fatalities and injuries.
 - Mean severity scores.

```
def aggregate_data(data): 1 usage • Reni Korueva
    aggregated = data.groupby(['Country', 'Year']).agg({
        'Accidents Reported': 'sum',
        'Fatalities': 'sum',
        'Injuries': 'sum',
        'Severity Score': 'mean'
    }).reset_index()
    return aggregated
```

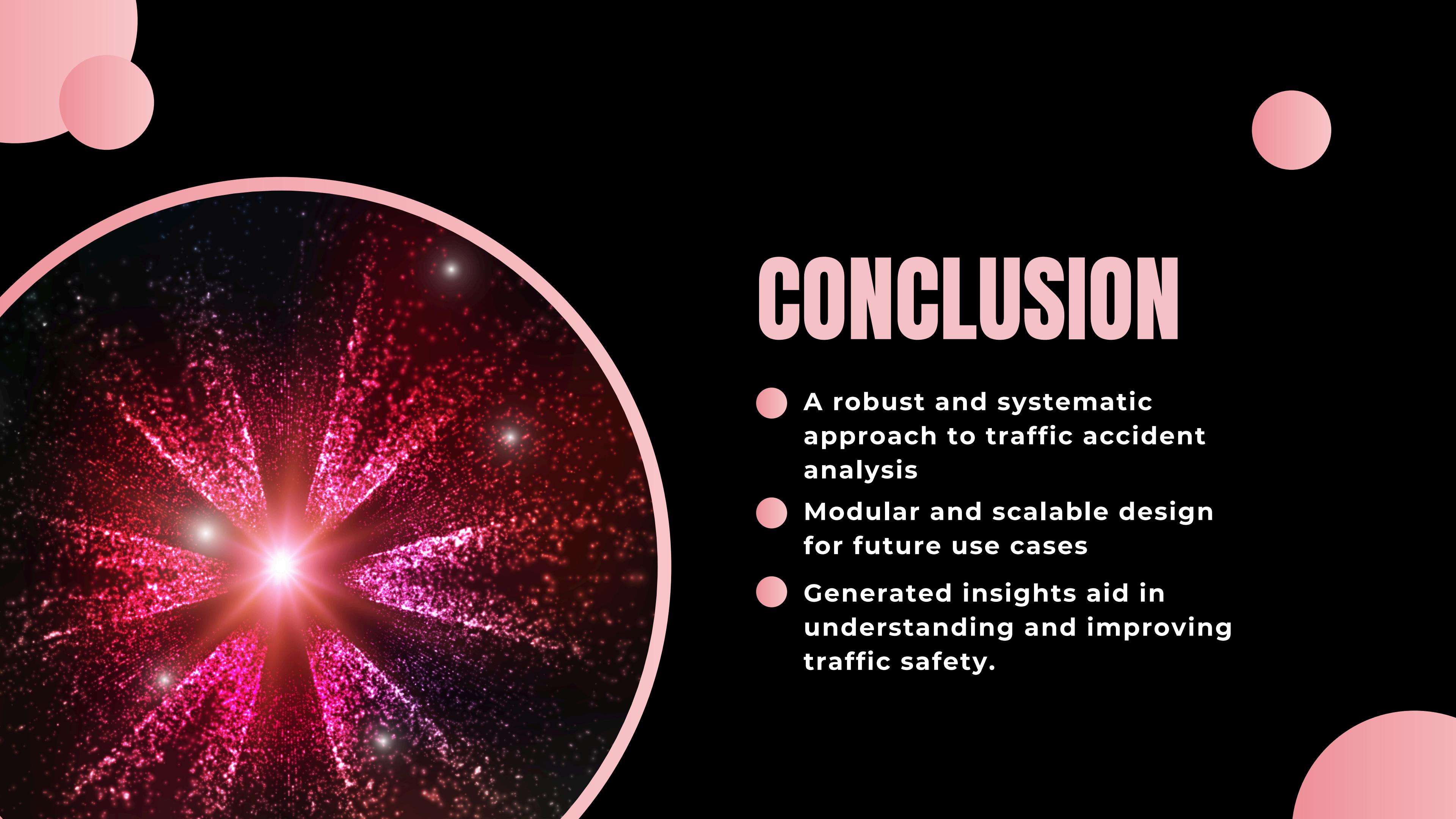
DATA VISUALIZATION



- **Trend Analysis** → Line plots for accident trends over time.
- **Comparative Analysis** → Bar charts for accident types and fatality rates.
- **Safety Measures Impact** → Demonstrates effectiveness of interventions.

SUMMARY OF ACHIEVEMENTS

- Prepared a clean and structured dataset.
- Generated insightful metrics: Severity Score, Fatality Rate, and Injury Rate.
- Created impactful visualizations.
- Automated the entire workflow for efficiency.



CONCLUSION

- A robust and systematic approach to traffic accident analysis
- Modular and scalable design for future use cases
- Generated insights aid in understanding and improving traffic safety.

THANK YOU!

Reni Korueva,
5MI3400592

Ana Ilieva,
5MI3400681

