**Data Wrangling and Analysis in Autonomous vehicles**

**Introduction**

Phase 2 of our project is dedicated to data wrangling and analysis, pivotal phases in refining the raw dataset for the development of autonomous vehicle systems. This phase entails the utilization of diverse data manipulation techniques, primarily in Python, to cleanse, transform, and delve into the dataset. Additionally, we operate under the premise of a scenario wherein the project aims to enhance the performance and safety of autonomous vehicles through advanced data analytics and modeling techniques.

**Objectives:**

1. Cleanse the Dataset: Address inconsistencies, errors, and missing values to ensure data integrity crucial for autonomous vehicle operations.

2. Explore Dataset Characteristics: Employ exploratory data analysis (EDA) to comprehend distributions and correlations within the dataset, essential for understanding driving patterns and environmental factors.

3. Feature Engineering: Engineer relevant features to augment model performance for precise navigation and decision-making in autonomous vehicles.

4. Document the Data Wrangling Process: Provide a comprehensive documentation of the data wrangling process, ensuring transparency and reproducibility in the development of autonomous vehicle systems.

**Dataset Description**

The dataset comprises sensor data collected from autonomous vehicles, encompassing information regarding vehicle dynamics, environmental conditions, and navigation decisions. Each entry in the dataset represents a snapshot of the vehicle's state and surroundings, forming the foundation for developing robust autonomous driving algorithms.

**Data Wrangling Techniques**

1. **Data Description**  
   **Head :** Displaying the first few rows of the dataset to get an initial overview.

**Tail :** Examining the last few rows of the dataset to ensure completeness.

**Info :** Obtaining information about the dataset structure, data types, and memory usage.

**Describe :** Generating descriptive statistics for numerical features to understand their distributions and central tendencies.

**Code:**

import pandas as pd

# Import the dataset

df = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

# Display first few rows

print("Head:")

print(df.head())

# Display last few rows

print("\nTail:")

print(df.tail())

# Display information about the dataset

print("\nInfo:")

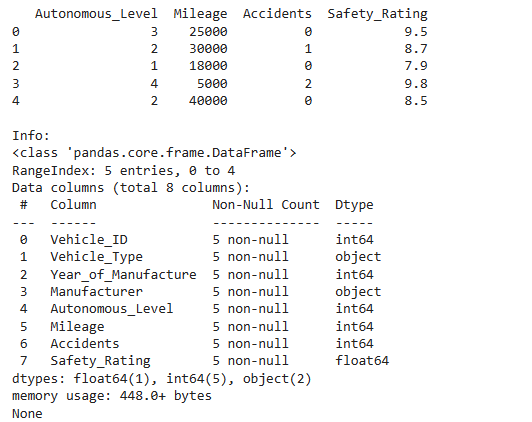
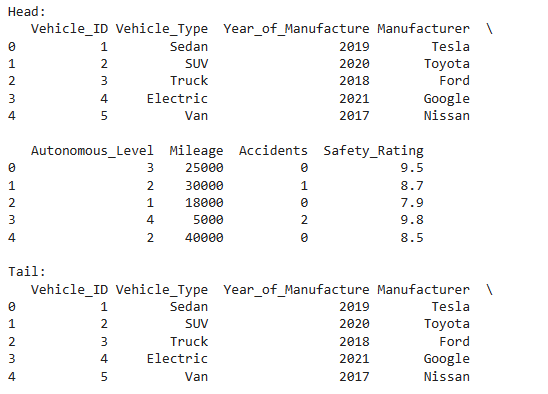
print(df.info())

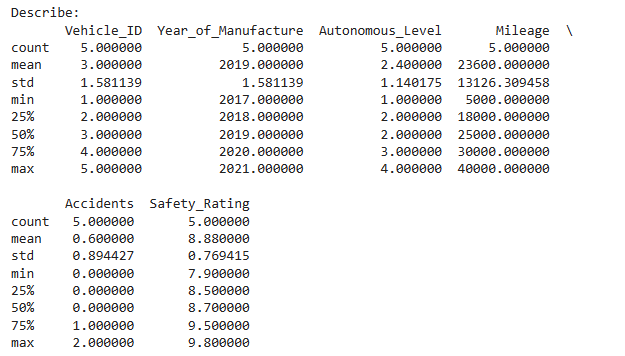
# Generate descriptive statistics for numerical features

print("\nDescribe:")

print(df.describe())

**Output:**





**2. Null Data Handling**

**Null Data Identification :** Identifying missing values in the dataset.

**Null Data Imputation :** Filling missing values with appropriate strategies.

**Null Data Removal :** Eliminating rows or columns with excessive missing values.

**Code:**

import pandas as pd

# Import the dataset

df = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Null\_data.csv')

# Display the original DataFrame

print("Original DataFrame:")

print(df)

# Null Data Identification

print("\nNull Data Identification:")

print(df.isnull())

# Null Data Imputation

print("\nNull Data Imputation:")

# Impute missing values in columns 'Mileage', 'Accidents', and 'Safety\_Rating' with mean

df['Mileage'] = df['Mileage'].fillna(df['Mileage'].mean())

df['Accidents'] = df['Accidents'].fillna(df['Accidents'].mean())

df['Safety\_Rating'] = df['Safety\_Rating'].fillna(df['Safety\_Rating'].mean())

# Impute missing values in columns 'Vehicle\_Type', 'Year\_of\_Manufacture', 'Manufacturer', and 'Autonomous\_Level' with specific values

df['Vehicle\_Type'] = df['Vehicle\_Type'].fillna('Unknown')

df['Year\_of\_Manufacture'] = df['Year\_of\_Manufacture'].fillna('Unknown')

df['Manufacturer'] = df['Manufacturer'].fillna('Unknown')

df['Autonomous\_Level'] = df['Autonomous\_Level'].fillna('Unknown')

print(df)

# Null Data Removal

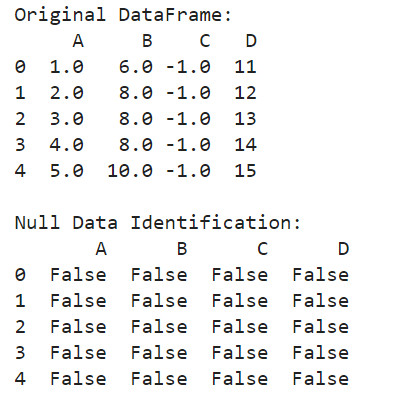
print("\nNull Data Removal:")

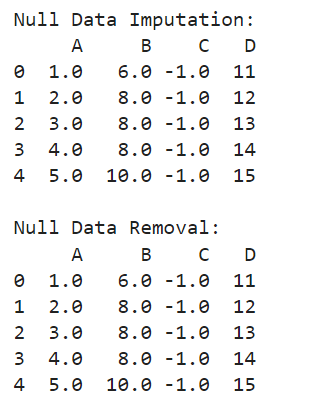
# Remove rows with any missing values

df\_cleaned = df.dropna()

print(df\_cleaned)

**Output:**





**3. Data Validation**

**Data Integrity Check :** Verifying data consistency and integrity to eliminate errors.

**Data Consistency Verification :** Ensuring data consistency across different columns or datasets.

**Code:**

import pandas as pd

#Import the dataset

df\_autonomous\_vehicles = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

# Data Integrity Check

print("Data Integrity Check:")

# Check data types of each column

print(df\_autonomous\_vehicles.dtypes)

# Check for missing values

print(df\_autonomous\_vehicles.isnull().sum())

# Data Consistency Verification

print("\nData Consistency Verification:")

# Verify consistency of Autonomous\_Level column (e.g., ensure values are within 1-5)

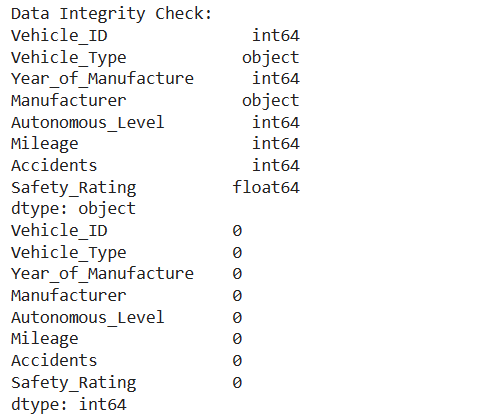
print("Unique Autonomous Levels:", df\_autonomous\_vehicles['Autonomous\_Level'].unique())

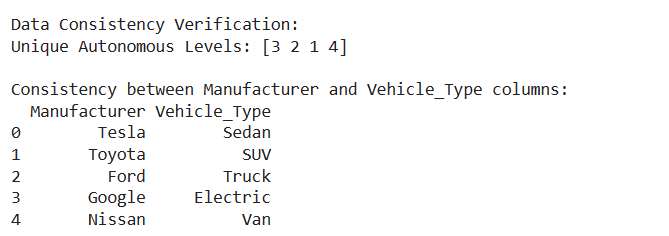
# Check for consistency between Manufacturer and Vehicle\_Type columns

print("\nConsistency between Manufacturer and Vehicle\_Type columns:")

print(df\_autonomous\_vehicles[['Manufacturer', 'Vehicle\_Type']])

**Output:**





**4. Data Reshaping**

**Reshaping Rows and Columns :** Transforming the dataset into a suitable format for analysis.

**Transposing Data :** Converting rows into columns and vice versa as needed.

**Code:**

import pandas as pd

# Import the dataset

df\_autonomous\_vehicles = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

# Display the original DataFrame

print("Original DataFrame:")

print(df\_autonomous\_vehicles)

# Reshaping Rows and Columns

print("\nReshaping Rows and Columns:")

# Transpose the DataFrame

df\_transposed = df\_autonomous\_vehicles.T

print(df\_transposed)

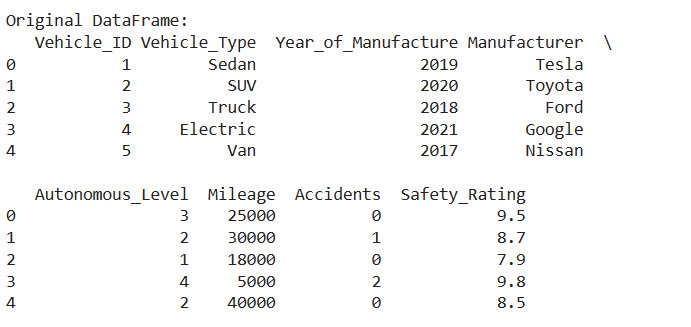
# Transposing Data

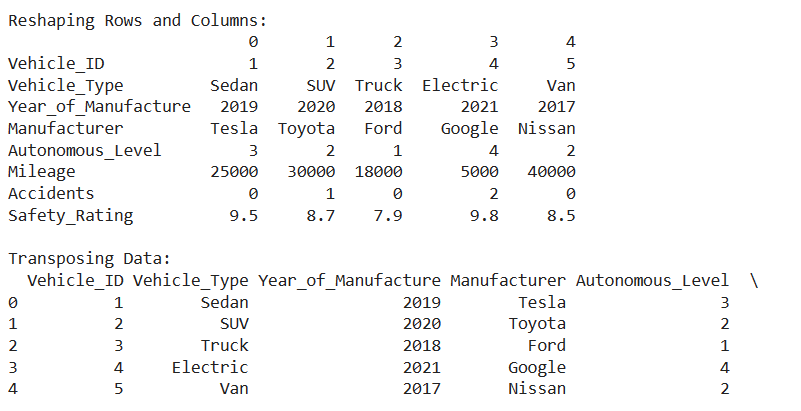
print("\nTransposing Data:")

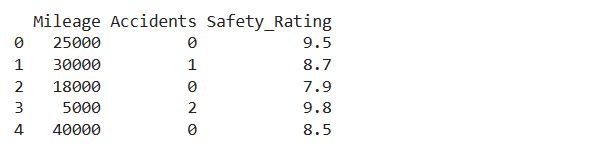
# Transpose the DataFrame back to its original shape

df\_original = df\_transposed.T

print(df\_original)

**Output:** 





**5. Data Merging**

**Combining Datasets :** Merging multiple datasets or data sources to enrich the information available for analysis.

**Joining Data :** Joining datasets based on common columns or keys.

**Code:**

import pandas as pd

#Import the datasets

df1 = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

df2 = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles1.csv')

# Display the original DataFrames

print("First DataFrame (df1):")

print(df1)

print("\nSecond DataFrame (df2):")

print(df2)

# Joining Data with specified suffixes

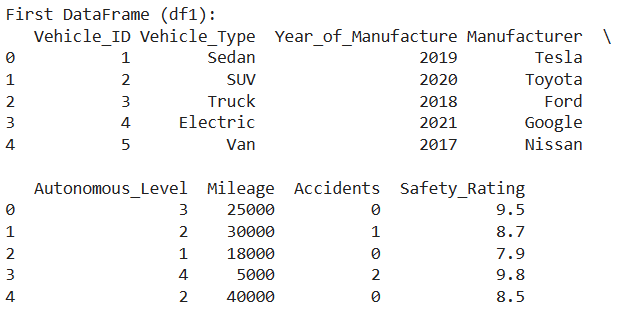
print("\nJoining Data:")

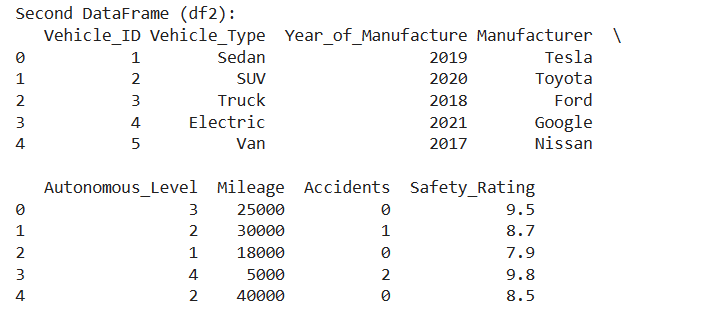
# Join the two DataFrames based on the common column 'Vehicle\_ID'

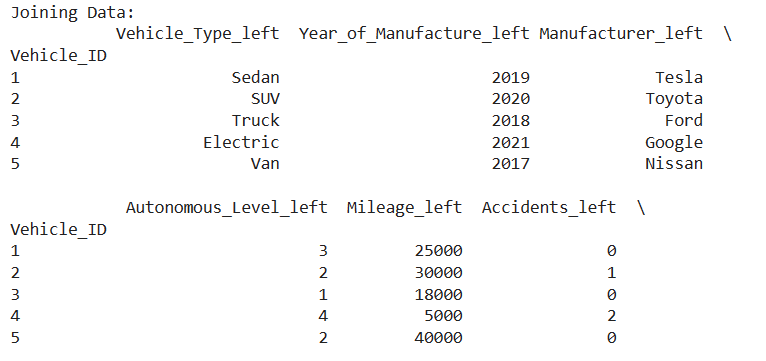
joined\_df = df1.set\_index('Vehicle\_ID').join(df2.set\_index('Vehicle\_ID'), how='outer', lsuffix='\_left', rsuffix='\_right')

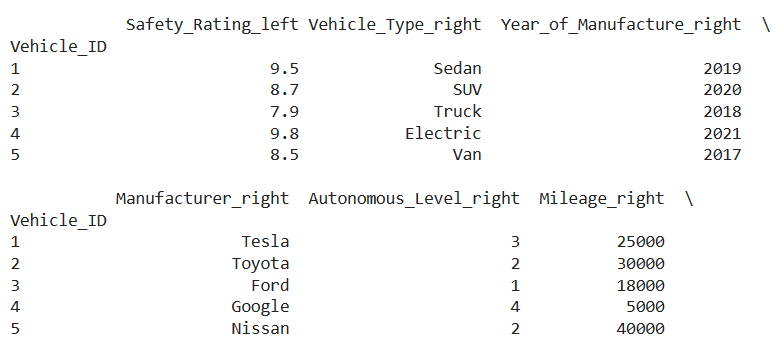
print(joined\_df)

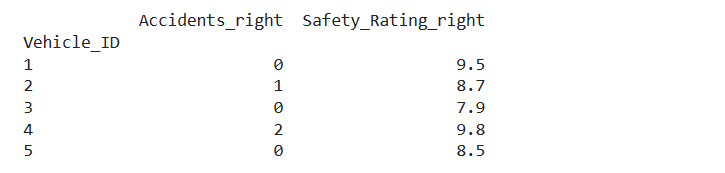
**Output:**











**6. Data Aggregation**

**Grouping Data :** Grouping dataset rows based on specific criteria.

**Aggregating Data :** Computing summary statistics for grouped data

**Code:**

import pandas as pd

#Import the dataset

df=pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

# Display the original DataFrame

print("Original DataFrame:")

print(df)

# Grouping Data

print("\nGrouping Data:")

# Group the DataFrame by 'Vehicle\_Type'

grouped\_df = df.groupby('Vehicle\_Type')

for group\_name, group\_data in grouped\_df:

    print("\nGroup:", group\_name)

    print(group\_data)

# Aggregating Data

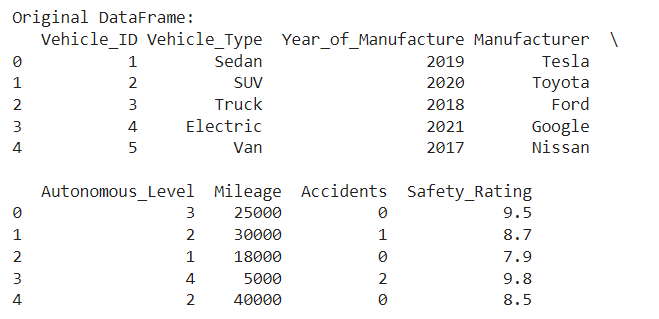
print("\nAggregating Data:")

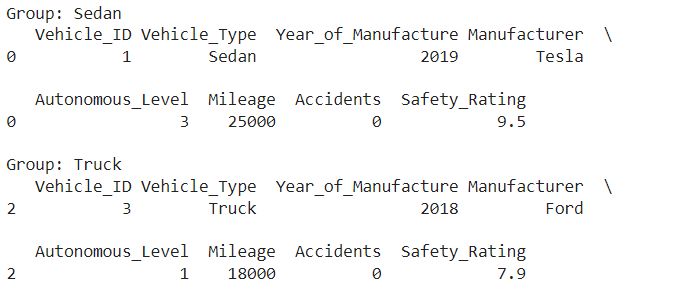
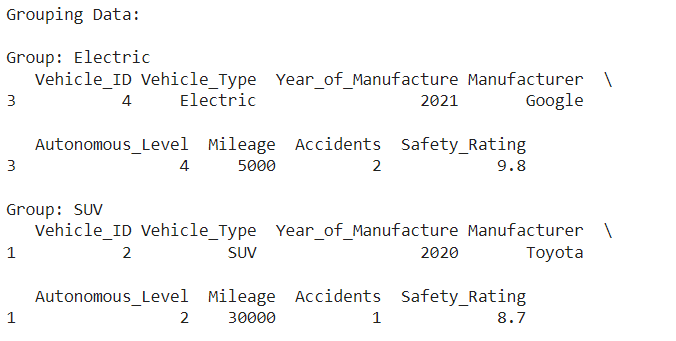
# Compute summary statistics for 'Mileage' grouped by 'Vehicle\_Type'

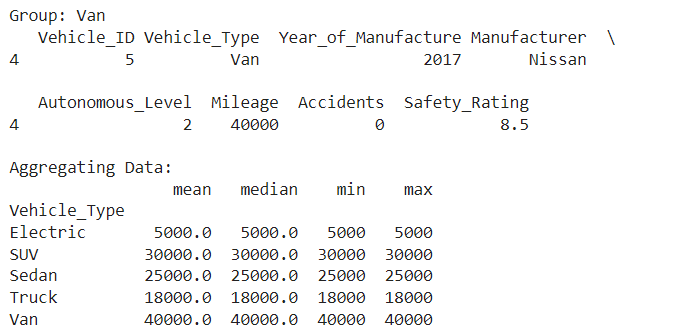
summary\_stats = grouped\_df['Mileage'].agg(['mean', 'median', 'min', 'max'])

print(summary\_stats)

**Output:**







**Data Analysis Techniques**

**7. Exploratory Data Analysis (EDA)**

**Univariate Analysis :** Analyzing individual variables to understand their distributions and characteristics.

**Bivariate Analysis :** Investigating relationships between pairs of variables to identify correlations and dependencies.

**Multivariate Analysis** : Exploring interactions among multiple variables to uncover complex patterns and trends.

**Code:**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Import the dataset

df = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

# Univariate Analysis

print("\nUnivariate Analysis:")

plt.figure(figsize=(8, 6))

sns.histplot(df['Mileage'], bins=10, kde=True)

plt.title('Distribution of Mileage')

plt.xlabel('Mileage (miles)')

plt.ylabel('Frequency')

plt.show()

plt.savefig('mileage\_distribution.png')

# Bivariate Analysis

print("\nBivariate Analysis:")

plt.figure(figsize=(8, 6))

sns.scatterplot(x='Mileage', y='Accidents', data=df)

plt.title('Relationship between Mileage and Accidents')

plt.xlabel('Mileage (miles)')

plt.ylabel('Number of Accidents')

plt.show()

plt.savefig('mileage\_accidents\_relationship.png')

# Multivariate Analysis

print("\nMultivariate Analysis:")

plt.figure(figsize=(8, 6))

numeric\_columns = df.select\_dtypes(include=['int64', 'float64']).columns

correlation\_matrix = df[numeric\_columns].corr()

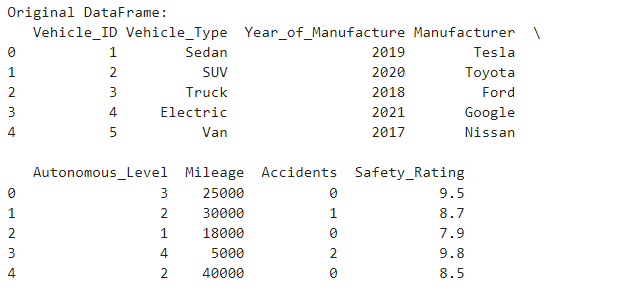
sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Heatmap')

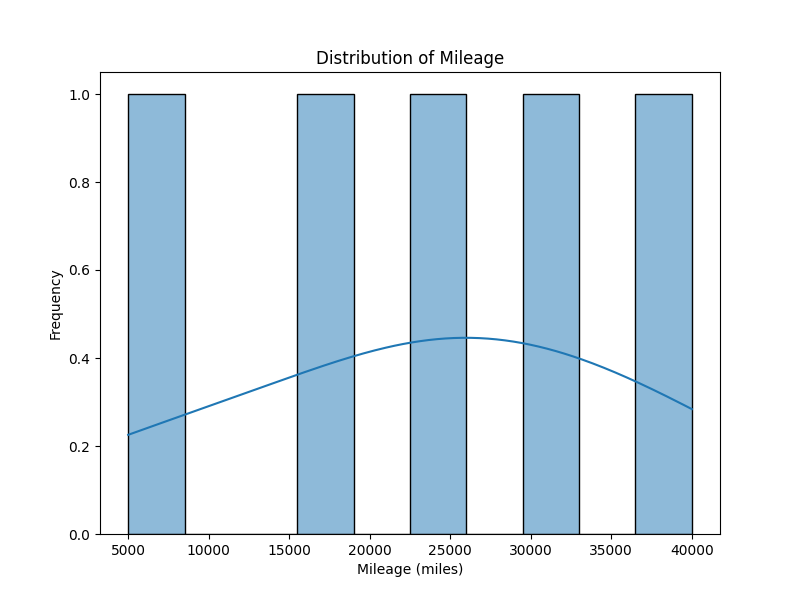
plt.show()

plt.savefig('correlation\_heatmap.png')

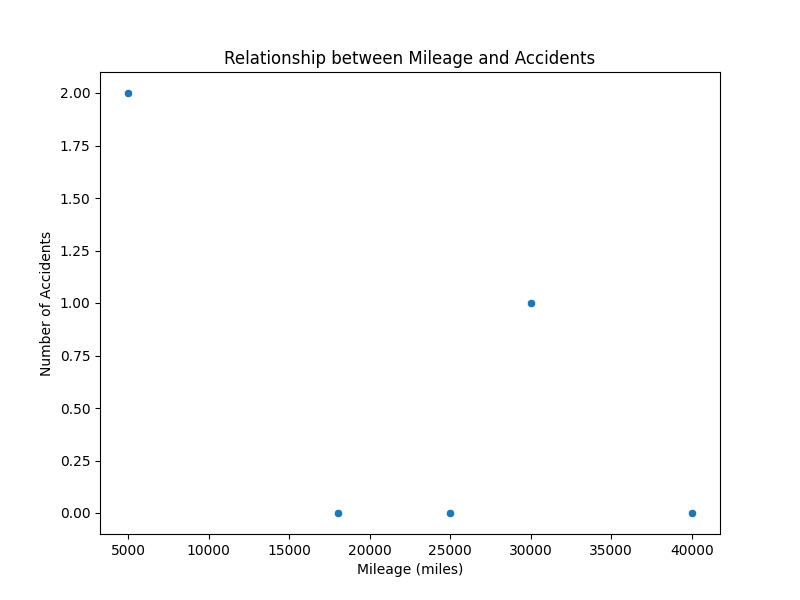
**Output:**



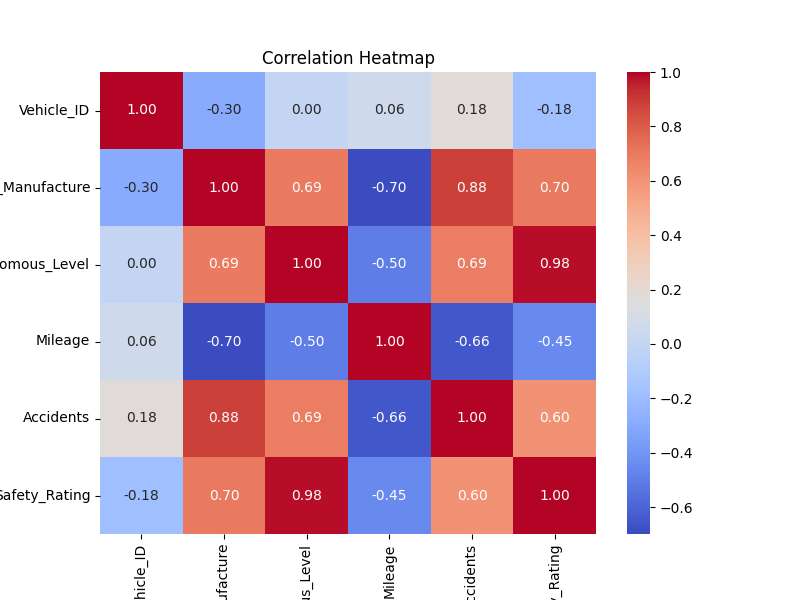
Univariate Analysis:

****

Bivariate Analysis:



Multivariate Analysis:



**Univariate analysis – Histogram:**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load the autonomous vehicles dataset

autonomous\_vehicles\_data = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

# Univariate Analysis - Histogram

print('\nUnivariate Analysis - Histogram')

sns.histplot(autonomous\_vehicles\_data['Mileage'], bins=20)

plt.title('Histogram of Mileage')

plt.xlabel('Mileage (miles)')

plt.ylabel('Frequency')

plt.show()

plt.savefig('Histogram of Mileage.png')

# Bivariate Analysis - Scatter plot

print('\nBivariate Analysis - Scatter plot')

sns.scatterplot(x='Mileage', y='Accidents', data=autonomous\_vehicles\_data)

plt.title('Scatter Plot between Mileage and Accidents')

plt.xlabel('Mileage (miles)')

plt.ylabel('Number of Accidents')

plt.show()

plt.savefig('Scatter Plot between Mileage and Accidents.png')

# Multivariate Analysis - Pair plot

print('\nMultivariate Analysis - Pair plot')

sns.pairplot(autonomous\_vehicles\_data)

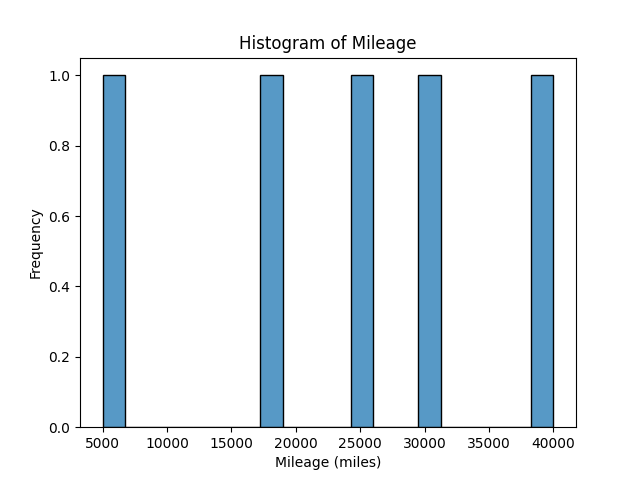
plt.title('Pair Plot of Autonomous Vehicles Data')

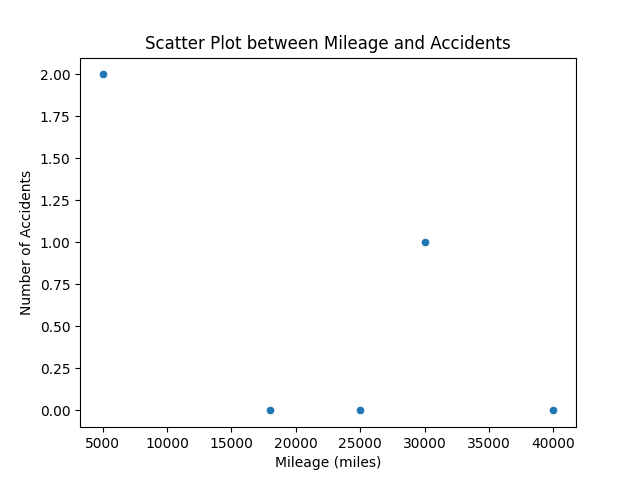
plt.show()

plt.savefig('Pair Plot of Autonomous Vehicles Data.png')

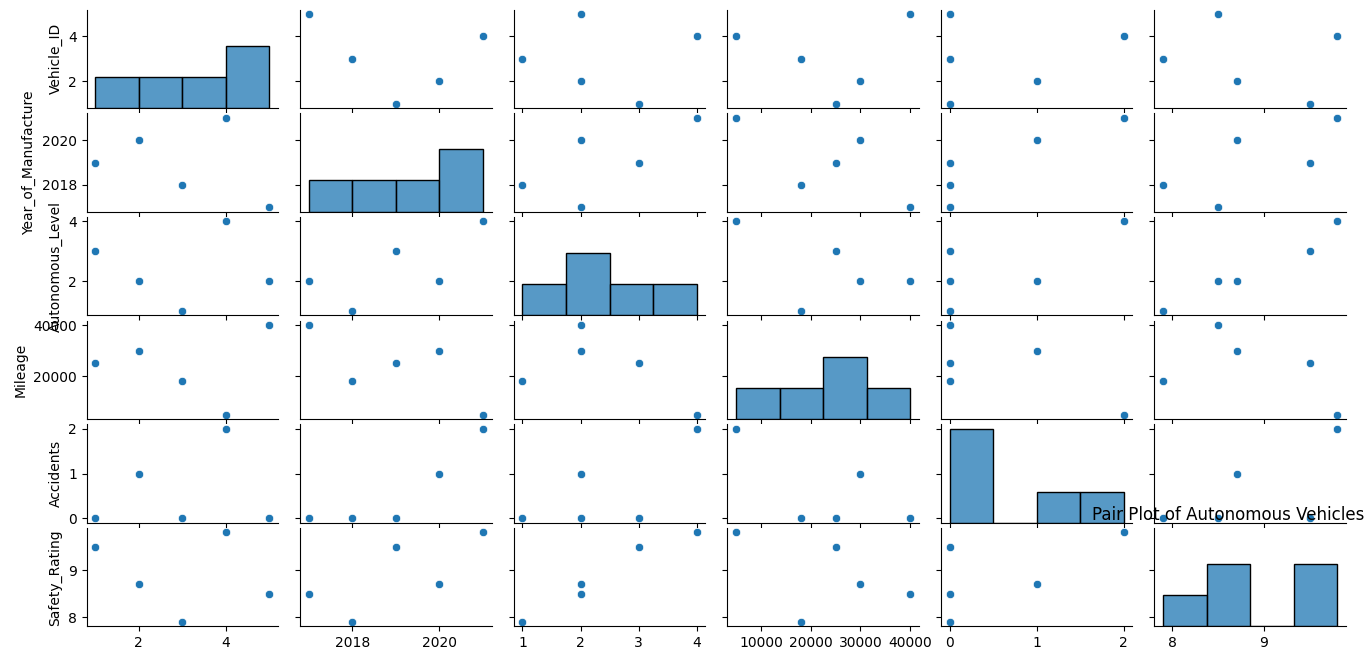
**Output:**

Univariate Analysis - Histogram



Bivariate Analysis - Scatter plot****

Multivariate Analysis - Pair plot



**8.Feature Engineering**

**Creating User Profiles:** Aggregating user interaction data to construct comprehensive user profiles capturing preferences and behaviours.

**Temporal Analysis:** Incorporating temporal features such as time of day or day of week to capture temporal trends in user behaviour.

**Content Embeddings:** Generating embeddings for content items to represent their characteristics and relationships.

import pandas as pd

from datetime import datetime

from gensim.models import Word2Vec

# Load the dataset

autonomous\_vehicles\_data = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

# Creating User Profiles

user\_profiles = autonomous\_vehicles\_data.groupby('Vehicle\_ID')['Accidents'].sum().reset\_index()

user\_profiles.rename(columns={'Accidents': 'Total\_Accidents'}, inplace=True)

print("User Profiles:")

print(user\_profiles)

# Temporal Analysis

current\_year = datetime.now().year

autonomous\_vehicles\_data['Year\_Manufacture\_Age'] = current\_year - autonomous\_vehicles\_data['Year\_of\_Manufacture']

print("\nTemporal Analysis:")

print(autonomous\_vehicles\_data.head())

# Content Embeddings (Not applicable for this dataset, but let's demonstrate with a dummy example)

content\_data = autonomous\_vehicles\_data[['Vehicle\_Type', 'Manufacturer', 'Autonomous\_Level']].copy()

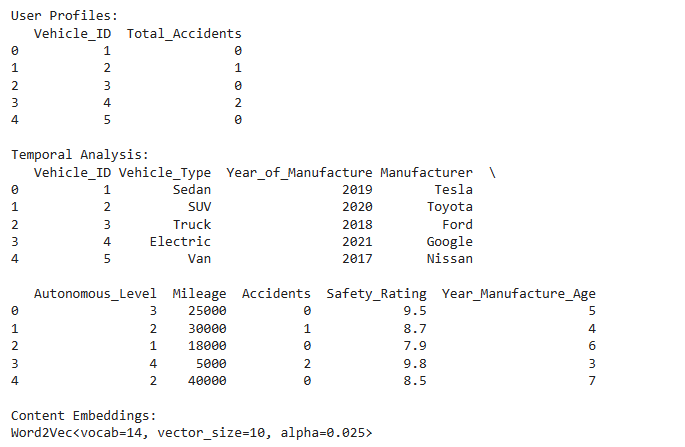
# Generate embeddings for each feature using Word2Vec

embeddings\_model = Word2Vec(sentences=content\_data.values.tolist(), vector\_size=10, window=5, min\_count=1, workers=4)

print("\nContent Embeddings:")

print(embeddings\_model)

**Output:**

****

**Assumed Scenario:**

The project aims to enhance the safety and efficiency of autonomous vehicles by leveraging historical data and advanced analytics techniques.

**Objective:**

Improve the performance and reliability of autonomous vehicles by utilizing historical interaction data and predictive analytics to anticipate and prevent potential issues.

**Target Audience:**

Developers, engineers, and stakeholders involved in the development and deployment of autonomous vehicle technologies.

**Conclusion:**

Phase 2 of the project focuses on data wrangling and analysis to harness the power of historical interaction data in optimizing the performance and safety of autonomous vehicles. By leveraging Python-based data manipulation techniques and predictive analytics, we aim to derive actionable insights that drive advancements in autonomous vehicle technology, ultimately leading to safer and more efficient transportation solutions.

**Code:**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from datetime import datetime

from gensim.models import Word2Vec

#1

df = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

# Display first few rows

print("Head:")

print(df.head())

# Display last few rows

print("\nTail:")

print(df.tail())

# Display information about the dataset

print("\nInfo:")

print(df.info())

# Generate descriptive statistics for numerical features

print("\nDescribe:")

print(df.describe())

#2

df = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Null\_data.csv')

# Display the original DataFrame

print("Original DataFrame:")

print(df)

# Null Data Identification

print("\nNull Data Identification:")

print(df.isnull())

# Null Data Imputation

print("\nNull Data Imputation:")

# Impute missing values in columns 'Mileage', 'Accidents', and 'Safety\_Rating' with mean

df['Mileage'] = df['Mileage'].fillna(df['Mileage'].mean())

df['Accidents'] = df['Accidents'].fillna(df['Accidents'].mean())

df['Safety\_Rating'] = df['Safety\_Rating'].fillna(df['Safety\_Rating'].mean())

# Impute missing values in columns 'Vehicle\_Type', 'Year\_of\_Manufacture', 'Manufacturer', and 'Autonomous\_Level' with specific values

df['Vehicle\_Type'] = df['Vehicle\_Type'].fillna('Unknown')

df['Year\_of\_Manufacture'] = df['Year\_of\_Manufacture'].fillna('Unknown')

df['Manufacturer'] = df['Manufacturer'].fillna('Unknown')

df['Autonomous\_Level'] = df['Autonomous\_Level'].fillna('Unknown')

print(df)

# Null Data Removal

print("\nNull Data Removal:")

# Remove rows with any missing values

df\_cleaned = df.dropna()

print(df\_cleaned)

#3

#Import the dataset

df\_autonomous\_vehicles = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

# Data Integrity Check

print("Data Integrity Check:")

# Check data types of each column

print(df\_autonomous\_vehicles.dtypes)

# Check for missing values

print(df\_autonomous\_vehicles.isnull().sum())

# Data Consistency Verification

print("\nData Consistency Verification:")

# Verify consistency of Autonomous\_Level column (e.g., ensure values are within 1-5)

print("Unique Autonomous Levels:", df\_autonomous\_vehicles['Autonomous\_Level'].unique())

# Check for consistency between Manufacturer and Vehicle\_Type columns

print("\nConsistency between Manufacturer and Vehicle\_Type columns:")

print(df\_autonomous\_vehicles[['Manufacturer', 'Vehicle\_Type']])

#4

# Import the dataset

df\_autonomous\_vehicles = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

# Display the original DataFrame

print("Original DataFrame:")

print(df\_autonomous\_vehicles)

# Reshaping Rows and Columns

print("\nReshaping Rows and Columns:")

# Transpose the DataFrame

df\_transposed = df\_autonomous\_vehicles.T

print(df\_transposed)

# Transposing Data

print("\nTransposing Data:")

# Transpose the DataFrame back to its original shape

df\_original = df\_transposed.T

print(df\_original)

#5

#Import the dataset

df1 = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

df2 = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles1.csv')

# Display the original DataFrames

print("First DataFrame (df1):")

print(df1)

print("\nSecond DataFrame (df2):")

print(df2)

# Joining Data with specified suffixes

print("\nJoining Data:")

# Join the two DataFrames based on the common column 'Vehicle\_ID'

joined\_df = df1.set\_index('Vehicle\_ID').join(df2.set\_index('Vehicle\_ID'), how='outer', lsuffix='\_left', rsuffix='\_right')

print(joined\_df)

#6

df=pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

# Display the original DataFrame

print("Original DataFrame:")

print(df)

# Grouping Data

print("\nGrouping Data:")

# Group the DataFrame by 'Vehicle\_Type'

grouped\_df = df.groupby('Vehicle\_Type')

for group\_name, group\_data in grouped\_df:

    print("\nGroup:", group\_name)

    print(group\_data)

# Aggregating Data

print("\nAggregating Data:")

# Compute summary statistics for 'Mileage' grouped by 'Vehicle\_Type'

summary\_stats = grouped\_df['Mileage'].agg(['mean', 'median', 'min', 'max'])

print(summary\_stats)

#7

# Import the dataset

df = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

# Univariate Analysis

print("\nUnivariate Analysis:")

plt.figure(figsize=(8, 6))

sns.histplot(df['Mileage'], bins=10, kde=True)

plt.title('Distribution of Mileage')

plt.xlabel('Mileage (miles)')

plt.ylabel('Frequency')

plt.show()

plt.savefig('mileage\_distribution.png')

# Bivariate Analysis

print("\nBivariate Analysis:")

plt.figure(figsize=(8, 6))

sns.scatterplot(x='Mileage', y='Accidents', data=df)

plt.title('Relationship between Mileage and Accidents')

plt.xlabel('Mileage (miles)')

plt.ylabel('Number of Accidents')

plt.show()

plt.savefig('mileage\_accidents\_relationship.png')

# Multivariate Analysis

print("\nMultivariate Analysis:")

plt.figure(figsize=(8, 6))

numeric\_columns = df.select\_dtypes(include=['int64', 'float64']).columns

correlation\_matrix = df[numeric\_columns].corr()

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Heatmap')

plt.show()

plt.savefig('correlation\_heatmap.png')

#8

autonomous\_vehicles\_data = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

# Univariate Analysis - Histogram

print('\nUnivariate Analysis - Histogram')

sns.histplot(autonomous\_vehicles\_data['Mileage'], bins=20)

plt.title('Histogram of Mileage')

plt.xlabel('Mileage (miles)')

plt.ylabel('Frequency')

plt.show()

plt.savefig('Histogram of Mileage.png')

# Bivariate Analysis - Scatter plot

print('\nBivariate Analysis - Scatter plot')

sns.scatterplot(x='Mileage', y='Accidents', data=autonomous\_vehicles\_data)

plt.title('Scatter Plot between Mileage and Accidents')

plt.xlabel('Mileage (miles)')

plt.ylabel('Number of Accidents')

plt.show()

plt.savefig('Scatter Plot between Mileage and Accidents.png')

# Multivariate Analysis - Pair plot

print('\nMultivariate Analysis - Pair plot')

sns.pairplot(autonomous\_vehicles\_data)

plt.title('Pair Plot of Autonomous Vehicles Data')

plt.show()

plt.savefig('Pair Plot of Autonomous Vehicles Data.png')

#9

# Load the dataset

autonomous\_vehicles\_data = pd.read\_csv('https://raw.githubusercontent.com/Tech-master1234/Naan-mudhalvan/main/Autonomous\_vehicles.csv')

# Creating User Profiles

user\_profiles = autonomous\_vehicles\_data.groupby('Vehicle\_ID')['Accidents'].sum().reset\_index()

user\_profiles.rename(columns={'Accidents': 'Total\_Accidents'}, inplace=True)

print("User Profiles:")

print(user\_profiles)

# Temporal Analysis

current\_year = datetime.now().year

autonomous\_vehicles\_data['Year\_Manufacture\_Age'] = current\_year - autonomous\_vehicles\_data['Year\_of\_Manufacture']

print("\nTemporal Analysis:")

print(autonomous\_vehicles\_data.head())

# Content Embeddings (Not applicable for this dataset, but let's demonstrate with a dummy example)

content\_data = autonomous\_vehicles\_data[['Vehicle\_Type', 'Manufacturer', 'Autonomous\_Level']].copy()

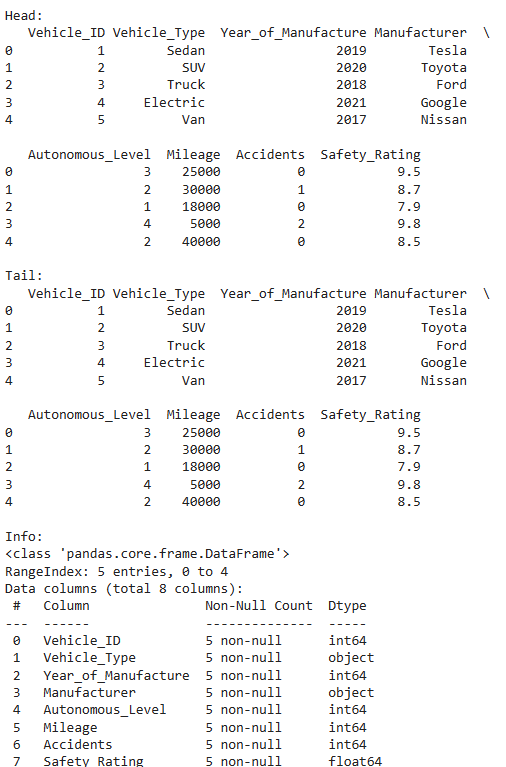
# Generate embeddings for each feature using Word2Vec

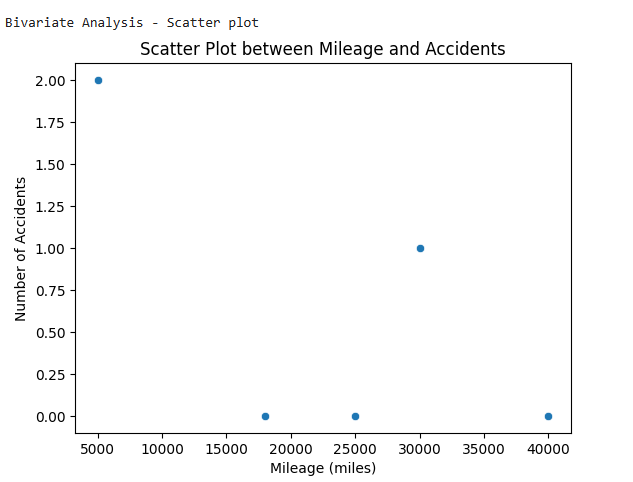
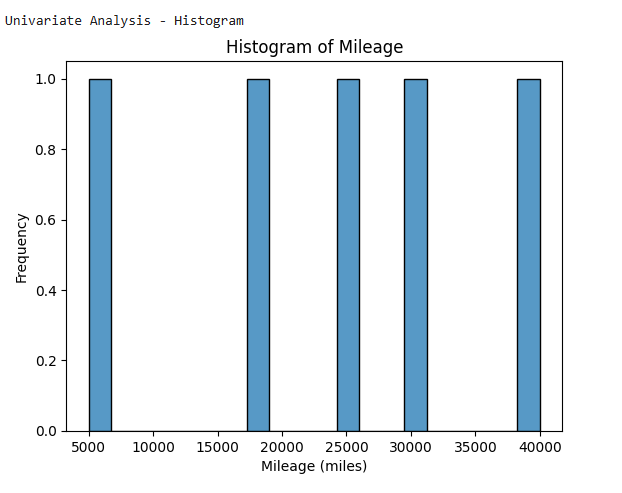
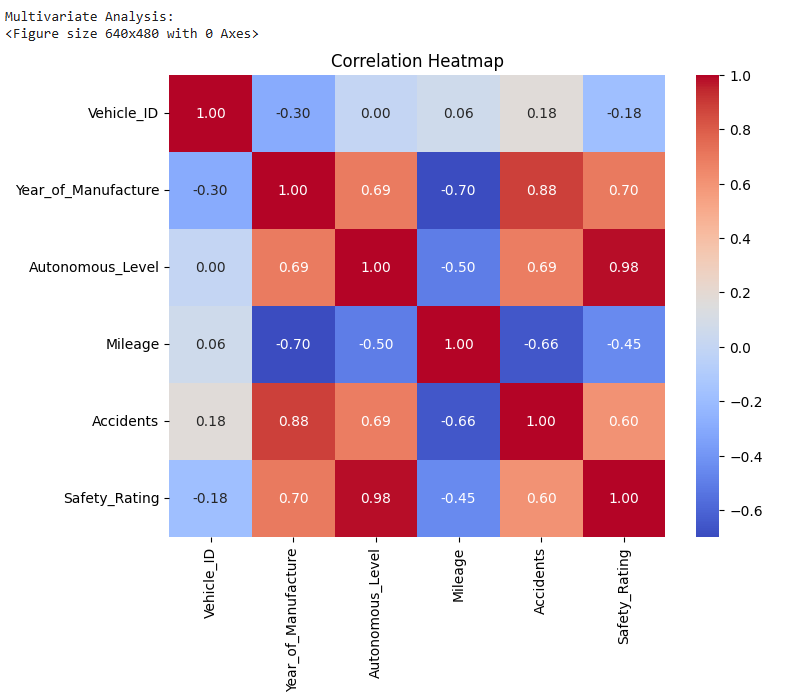
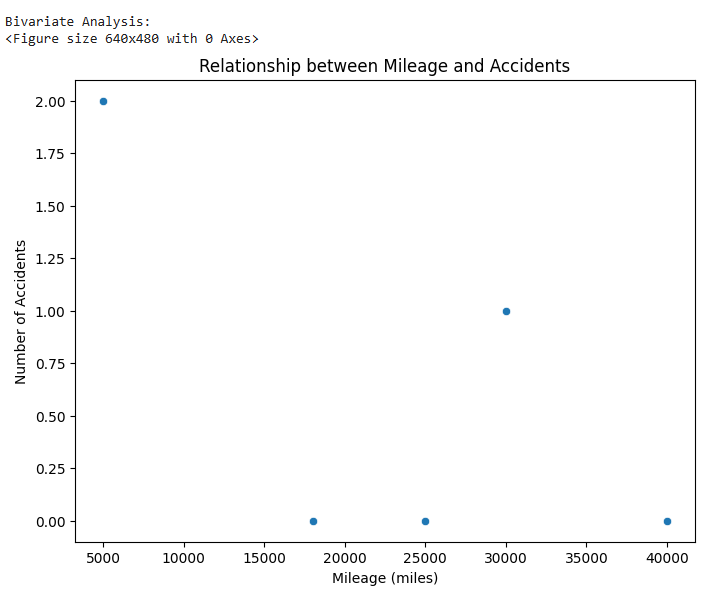
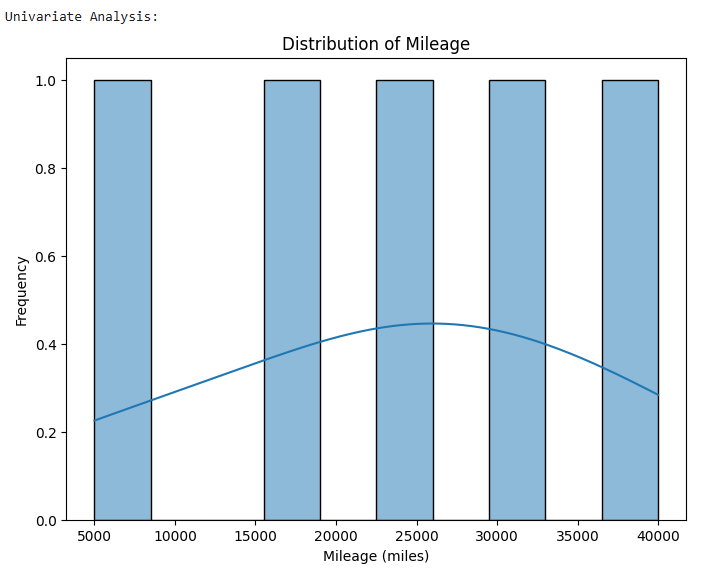
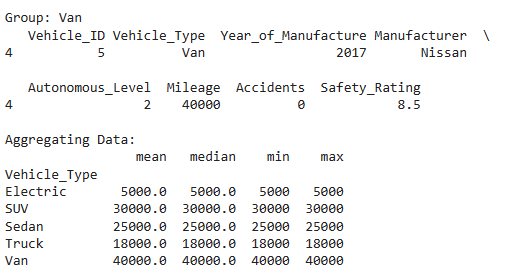
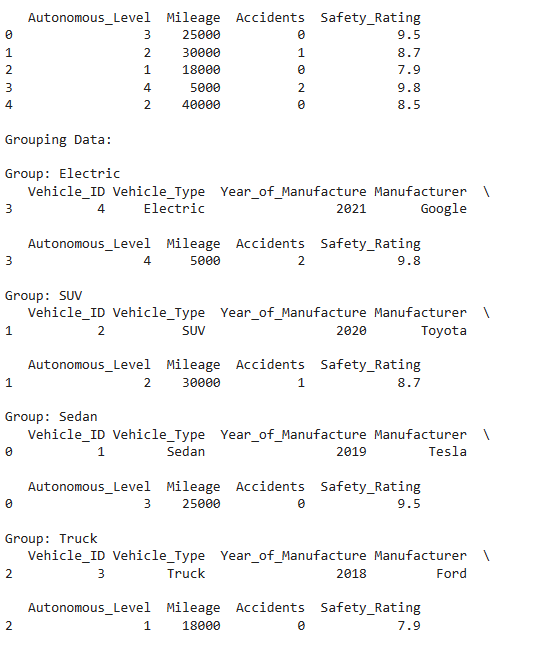
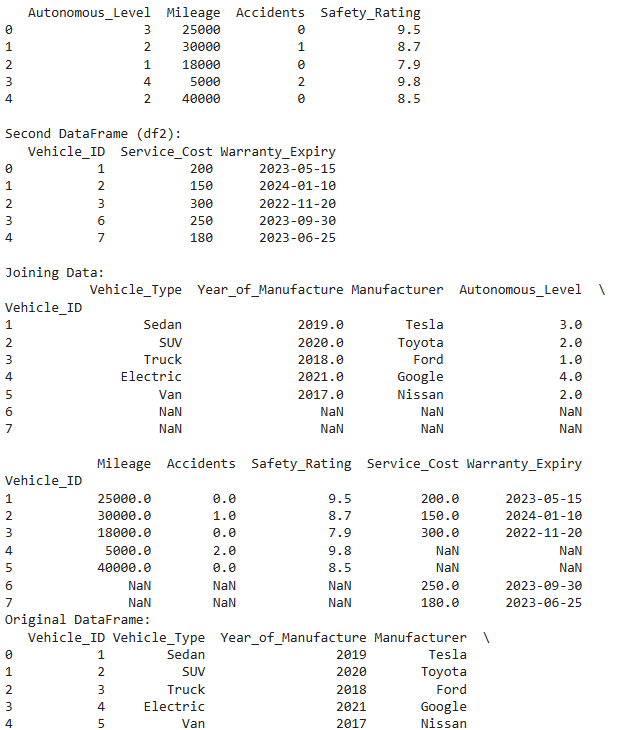
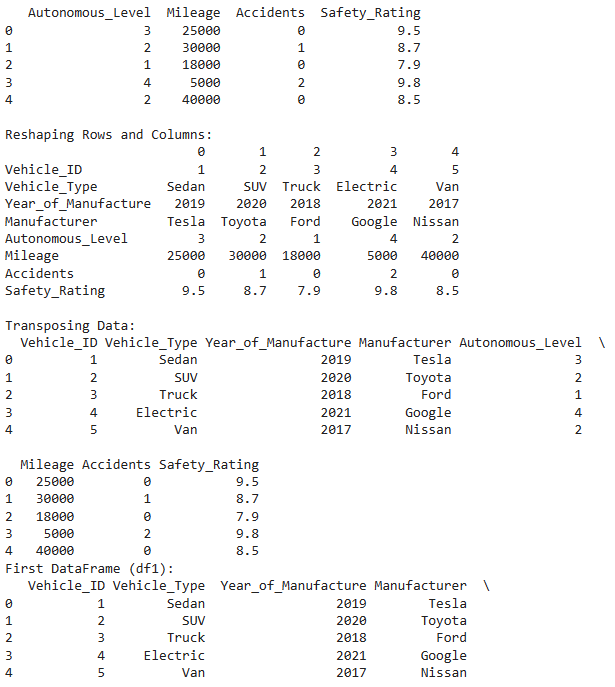
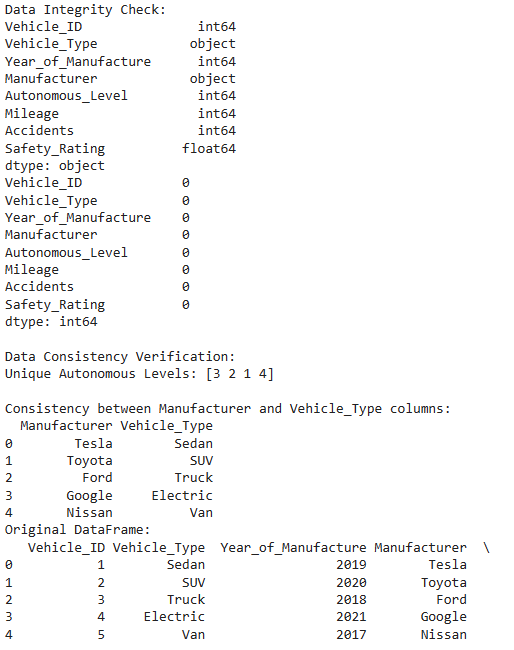
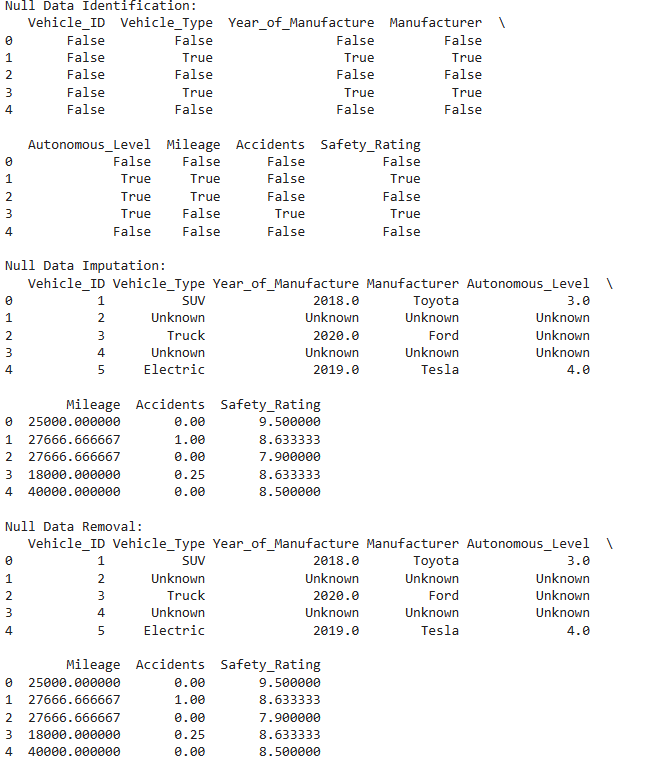
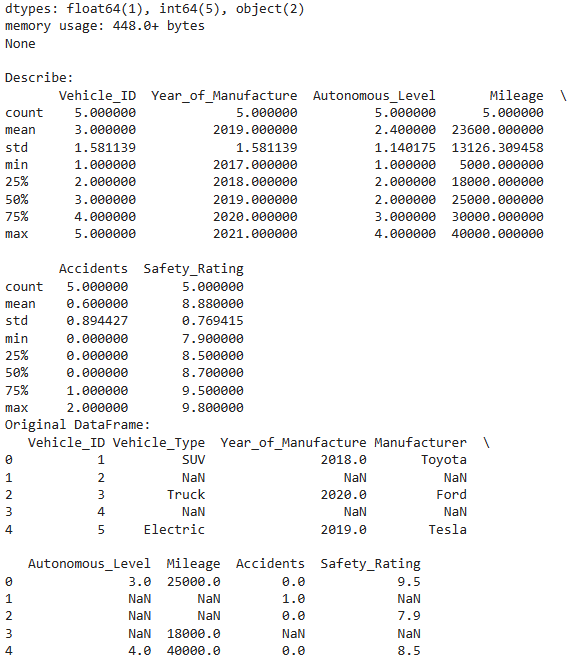
embeddings\_model = Word2Vec(sentences=content\_data.values.tolist(), vector\_size=10, window=5, min\_count=1, workers=4)

print("\nContent Embeddings:")

print(embeddings\_model)

**Output:**

****

****

C:\Users\manoj\Pictures\Screenshots\Screenshot 2024-05-05 091751.png

