NAAN MUDHALVAN
PROJECT PHASE – II

NAME : M.SHOBEYA

DOMAIN : ARTIFICIAL INTELLIGENCE

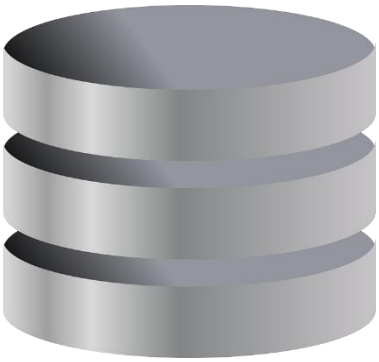
TOPIC : DEVELOPMENT OF AUTONOMOUS VEHICLES

DEPARTMENT : COMPUTER SCIENCE

COLLEGE : 8201 – ARJCET

DATA WRANGLING AND ANALYSIS

- ✓ INTRODUCTION
- ✓ OBJECTIVES
- ✓ DATASET DESCRIPTION
- ✓ DATA WRANGLING TECHNIQUES
- ✓ ASSUMED SCENARIO
- ✓ CONCLUSION



INTRODUCTION

In this phase of the project, we are going to perform data wran gling and analysis. To perform data wrangling we need some kind of dataset so I had taken my dataset as 2019 AV disengagement reports. We are going to perform various data manipulation techniques using python to clean, transform and explore the dataset.

OBJECTIVES

- Cleanse the dataset by removing inconsistent data, errors, and missing values

- Explore the dataset's characteristics through exploratory data analysis(EDA) to understand distributions and correlations.
- Perform data validation and data aggregation and visualization
- Provide feature engineering and temporal analysis.

DATASET DESCRIPTION

Dataset Title: 2019 Autonomous Disengagement Reports

Description:

The dataset contains records of autonomous vehicle disengagements reported by companies testing

autonomous vehicles on public roads in California during the year 2019. A disengagement occurs when a human driver intervenes and takes control of the vehicle due to safety concerns or when the autonomous system fails to operate as intended. Each record typically includes information such as the date, time, location, reason for disengagement, duration of human control, vehicle speed, and other relevant details.

Data Sources:

California Department of Motor Vehicles (DMV) reports

Company disclosures and submissions

Variables:

- *Date and Time*: Timestamp indicating when the disengagement occurred.
- *Location*: Geographic coordinates or description of where the disengagement occurred.
- *Company*: Name of the company conducting the autonomous vehicle testing.
- *Vehicle ID*: Identifier for the autonomous vehicle involved.
- *Reason for Disengagement:* Description of why the human driver intervened (e.g., safety concern, system failure).

- *Duration of Human Control:* Length of time the human driver maintained control of the vehicle.
- *Vehicle Speed:* Speed of the vehicle at the time of disengagement.
- *Environmental Conditions:* Weather conditions, road conditions, lighting, etc., at the time of disengagement.
- *Outcome:* Any consequences or incidents resulting from the disengagement.
- *Additional Notes:* Any other relevant information or commentary provided by the reporting company.

Potential Uses:

- ✦ Analyzing trends in autonomous vehicle performance and safety over time.
- ✦ Identifying common reasons for disengagements and areas for improvement in autonomous systems.
- ✦ Comparing disengagement rates and performance across different companies and vehicle models.

Limitations:

- The dataset may not capture all disengagement events, as reporting requirements and definitions vary between companies and jurisdictions.

- Disengagement reports may be subject to biases or inaccuracies in reporting.

- Some information, such as proprietary technical details or sensitive incident reports, may be redacted or unavailable.

Access and Availability:

The dataset may be available through public records requests to the California DMV or through disclosures made by individual companies conducting autonomous vehicle testing. Access may be subject to legal and privacy restrictions.

This description provides an overview of what the dataset contains, where it comes from, what variables are included, how it might be used, its limitations, and how it can be accessed.

DATA WRANGLING TECHNIQUES

1.DATA DESCRIPTION

head(): Returns the first few rows of the dataset, providing a quick glimpse of its structure and contents.

CODE

```
df.head()
```

OUTPUT

```
[7]: df.head()
```

[7]:

	Manufacturer	Permit Number	DATE	VIN NUMBER	VEHICLE IS CAPABLE OF OPERATING WITHOUT A DRIVER\n(Yes or No)	DRIVER PRESENT\n(Yes or No)	DISENGAGEMENT INITIATED BY\n(AV System, Test Driver, Remote Operator, or Passenger)	DISENGAGEMENT\nLOCATION\n(Interstate, Freeway, Highway, Rural Road, Street, or Parking Facility)	DESCRIPTION OF FACTS CAUSING DISENGAGEMENT
0	Ambarella Corp.	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected result from the path planner in the...
1	Ambarella Corp.	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected result from the radar based percept...
2	Ambarella Corp.	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected result from the path planner in the...
3	Ambarella Corp.	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected result from the GPS system in the g...
4	Ambarella Corp.	AVT053	3/15/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected result from the GPS system in the g...

tail(): Retrieves the last few rows of the dataset, useful for observing the concluding entries or checking data integrity at the end.

CODE

```
df.tail()
```

OUTPUT

```
[8]: df.tail()
```

[8]:

	Manufacturer	Permit Number	DATE	VIN NUMBER	VEHICLE IS CAPABLE OF OPERATING WITHOUT A DRIVER\n(Yes or No)	DRIVER PRESENT\n(Yes or No)	DISENGAGEMENT INITIATED BY\n(AV System, Test Driver, Remote Operator, or Passenger)	DISENGAGEMENT\nLOCATION\n(Interstate, Freeway, Highway, Rural Road, Street, or Parking Facility)	DESCRIPTION FACTS CAUSI DISENGAGEME
449	ThorDrive, Inc.	AVT064	4/29/2019	1FTYE1CM8JKA52066	No	Yes	Test Driver	Downtown street	Construction the w
450	ThorDrive, Inc.	AVT064	05-01-2019	1FTYE1CM8JKA52066	No	Yes	Test Driver	Downtown street	Construction the w
451	ThorDrive, Inc.	AVT064	05-06-2019	1FTYE1CM8JKA52066	No	Yes	Test Driver	Downtown street	Reckless driv road user t came from bel
452	ThorDrive, Inc.	AVT064	05-08-2019	1FTYE1CM8JKA52066	No	Yes	Test Driver	Downtown street	Incorrect behav prediction c bicycl
453	ThorDrive, Inc.	AVT064	6/24/2019	1FTYE1CM8JKA52066	No	Yes	Test Driver	Downtown street	Hardw connect dropping due t bu

info(): Presents essential information about the dataset, such as data types, memory usage, and the presence of missing values, aiding in initial data exploration and understanding.

CODE

```
df.info()
```

OUTPUT

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 454 entries, 0 to 453
Data columns (total 11 columns):
 #   Column                                                                                Non-Null Count  Dtype
---  ------                                                                                --------------  -----
 0   Manufacturer                                                                          454 non-null    object
 1   Permit Number                                                                         454 non-null    object
 2   DATE                                                                                  454 non-null    object
 3   VIN NUMBER                                                                            454 non-null    object
 4   VEHICLE IS CAPABLE OF OPERATING WITHOUT A DRIVER
(Yes or No)                                    454 non-null    object
 5   DRIVER PRESENT
(Yes or No)                                                                  454 non-null    object
 6   DISENGAGEMENT INITIATED BY
(AV System, Test Driver, Remote Operator, or Passenger)             454 non-null    object
 7   DISENGAGEMENT
LOCATION
(Interstate, Freeway, Highway, Rural Road, Street, or Parking Facility)  454 non-null    object
 8   DESCRIPTION OF FACTS CAUSING DISENGAGEMENT                                            454 non-null    object
 9   Unnamed: 9                                                                            24 non-null     object
 10  Unnamed: 10                                                                           1 non-null      object
dtypes: object(11)
memory usage: 39.1+ KB
```

describe(): Generates descriptive statistics summarizing the central tendency, dispersion, and shape of the dataset's numerical variables, facilitating a deeper insight into its numerical attributes.

CODE

```
df.describe()
```

OUTPUT

```
df.describe()
```

	Manufacturer	Permit Number	DATE	VIN NUMBER	VEHICLE IS CAPABLE OF OPERATING WITHOUT A DRIVER\n(Yes or No)	DRIVER PRESENT\n(Yes or No)	DISENGAGEMENT INITIATED BY\n(AV System, Test Driver, Remote Operator, or Passenger)	DISENGAGEMENT\nLOCATION\n(Interstate, Freeway, Highway, Rural Road, Street, or Parking Facility)	DESCRIPTI FACTS CA DISENGAGE
count	454	454	454	454	454	454	454	454	
unique	8	8	144	15	2	1	4	4	
top	Intel Corporation	AVT052	06-07-2018	3FA6P0LU4HR195512	Yes	Yes	Test Driver	Street	So Discre
freq	165	165	28	154	274	454	393	375	

2. NULL DATA HANDLING

isnull().sum()

Calculates the total count of missing values for each column in the dataset, providing insight into the extent of missing data.

CODE

df.isnull().sum()

OUTPUT

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

```
Manufacturer                                                                    0
Permit Number                                                                   0
DATE                                                                            0
VIN NUMBER                                                                      0
VEHICLE IS CAPABLE OF OPERATING WITHOUT A DRIVER\n(Yes or No)                   0
DRIVER PRESENT\n(Yes or No)                                                     0
DISENGAGEMENT INITIATED BY\n(AV System, Test Driver, Remote Operator, or Passenger)    0
DISENGAGEMENT\nLOCATION\n(Interstate, Freeway, Highway, Rural Road, Street, or Parking Facility)    0
DESCRIPTION OF FACTS CAUSING DISENGAGEMENT                                      0
Unnamed: 9                                                                    430
Unnamed: 10                                                                   453
dtype: int64
```

dropna(axis=0)

Removes rows from the dataset where any element is missing (NaN/null), aiding in data cleaning by eliminating incomplete records.

CODE

df.dropna(axis=0)

OUTPUT

```
df.dropna(axis=0)
```

	Manufacturer	Permit Number	DATE	VIN NUMBER	VEHICLE IS CAPABLE OF OPERATING WITHOUT A DRIVER\n(Yes or No)	DRIVER PRESENT\n(Yes or No)	DISENGAGEMENT INITIATED BY\n(AV System, Test Driver, Remote Operator, or Passenger)	DISENGAGEMENT\nLOCATION\n(Interstate, Freeway, Highway, Rural Road, Street, or Parking Facility)	DESCRIPTION FACTS CAUSI DISENGAGEME
239	Gatik AI Inc.	AVT054	01-10-2019	52CG2DGA7J0017503	No	Yes	Test Driver	Street	Reckless Age Road U

dropna(axis=1)

Removes columns from the dataset where any element is missing (NaN/null), facilitating data cleaning by eliminating incomplete features or attributes.

CODE

```
df.dropna(axis=1)
```

OUTPUT

```
df.dropna(axis=1)
```

	Manufacturer	Permit Number	DATE	VIN NUMBER	VEHICLE IS CAPABLE OF OPERATING WITHOUT A DRIVER\n(Yes or No)	DRIVER PRESENT\n(Yes or No)	DISENGAGEMENT INITIATED BY\n(AV System, Test Driver, Remote Operator, or Passenger)	DISENGAGEMENT\nLOCATION\n(Interstate, Freeway, Highway, Rural Road, Street, or Parking Facility)	DESCRIPTION FACTS CAUS DISENGAGEM
0	Ambarella Corp.	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected re from the p planner in t
1	Ambarella Corp.	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected re from the r based perce
2	Ambarella Corp.	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected re from the p planner in t
3	Ambarella Corp.	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected re from the system in the
4	Ambarella Corp.	AVT053	3/15/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected re from the system in the
...	...	...	...	...	...	...	...	...	
									Constructio

dropna(how='any')

Removes rows or columns from the dataset where any element is missing (NaN/null), helping to clean the dataset by eliminating incomplete observations or features.

CODE

```
df.dropna(how='any')
```

OUTPUT

```
df.dropna(how='any')
```

	Manufacturer	Permit Number	DATE	VIN NUMBER	VEHICLE IS CAPABLE OF OPERATING WITHOUT A DRIVER\n(Yes or No)	DRIVER PRESENT\n(Yes or No)	DISENGAGEMENT INITIATED BY\n(AV System, Test Driver, Remote Operator, or Passenger)	DISENGAGEMENT\nLOCATION\n(Interstate, Freeway, Highway, Rural Road, Street, or Parking Facility)	DESCRIPTION FACTS CAUSI DISENGAGEME
239	Gatik AI Inc.	AVT054	01-10-2019	52CG2DGA7J0017503	No	Yes	Test Driver	Street	Reckless Age Road U

dropna(how='all')

Removes rows or columns from the dataset where all elements are missing (NaN/null), useful for eliminating

entirely empty records or features during data cleaning.

CODE

```
df.dropna(how='all')
```

OUTPUT

```
[15]: df.dropna(how='all')
```

[15]:

	Manufacturer	Permit Number	DATE	VIN NUMBER	VEHICLE IS CAPABLE OF OPERATING WITHOUT A DRIVER\n(Yes or No)	DRIVER PRESENT\n(Yes or No)	DISENGAGEMENT INITIATED BY\n(AV System, Test Driver, Remote Operator, or Passenger)	DISENGAGEMENT\nLOCATION\n(Interstate, Freeway, Highway, Rural Road, Street, or Parking Facility)	DESCRIPTION FACTS CAUS DISENGAGEM
0	Ambarella Corp.	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected re from the planner in t
1	Ambarella Corp.	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected re from the r based perce
2	Ambarella Corp.	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected re from the planner in t
3	Ambarella Corp.	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected re from the system in the
4	Ambarella Corp.	AVT053	3/15/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected re from the system in the
...	...	...	...	...	...	...	...	...	

Constructio

dropna(thresh=50)

Removes rows from the dataset that have less than 50 non-null values, ensuring that only rows with at least 50 non-null values are retained, which can

be useful for filtering out incomplete records during data cleaning.

CODE

```
df.dropna(thresh=50)
```

OUTPUT

```
df.dropna(thresh=50)
```

Manufacturer	Permit Number	DATE	VIN NUMBER	VEHICLE IS CAPABLE OF OPERATING WITHOUT A DRIVER\n(Yes or No)	DRIVER PRESENT\n(Yes or No)	DISENGAGEMENT INITIATED BY\n(AV System, Test Driver, Remote Operator, or Passenger)	DISENGAGEMENT\nLOCATION\n(Interstate, Freeway, Highway, Rural Road, Street, or Parking Facility)	DESCRIPTION OF FACTS CAUSING DISENGAGEMENT	Unnamed: 9	Un

```
df.dropna(axis=1,thresh=20)
```

	Manufacturer	Permit Number	DATE	VIN NUMBER	VEHICLE IS CAPABLE OF OPERATING WITHOUT A DRIVER\n(Yes or No)	DRIVER PRESENT\n(Yes or No)	DISENGAGEMENT INITIATED BY\n(AV System, Test Driver, Remote Operator, or Passenger)	DISENGAGEMENT\nLOCATION\n(Interstate, Freeway, Highway, Rural Road, Street, or Parking Facility)	DESCRIPTION FACTS CAUS DISENGAGEM
0	Ambarella Corp.	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected r from the planner in t
1	Ambarella Corp.	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected r from the r based perc
2	Ambarella	AVT053	3/14/2018	3LN6L5MU7HR609845	No	Yes	test driver	street	Unexpected r from the

3.DATA VALIDATION

Data validation: Data validation is the process of ensuring that data is accurate, reliable, and suitable for its intended use. It involves checking data against predefined rules, constraints, or

standards to identify errors, inconsistencies, or anomalies and correcting them to maintain data quality.

Data consistency verification: Data consistency verification is a subset of data validation focused specifically on ensuring that data across different sources, systems, or components is synchronized and coherent. It involves comparing data elements or records to detect discrepancies, ensuring uniformity and reliability across the dataset.

unique()

unique(): In data validation, the unique() function identifies the distinct values present in a dataset's column,

providing a quick overview of the data's diversity and aiding in the validation process by ensuring data integrity and identifying potential duplicates or anomalies.

CODE

df[‘column_name’].unique()

OUTPUT

```
df['Manufacturer'].unique()
```

```
array(['Ambarella Corp.', 'Apex.Ai, Inc.', 'Box Bot Inc.',
       'DiDi Research America, LLC', 'Gatik AI Inc.', 'Intel Corporation',
       'RIDECELL INC', 'ThorDrive, Inc.'], dtype=object)
```

for column 1

```
]: df['Permit Number'].unique()
```

```
]: array(['AVT053', 'AVT051', 'AVT057', 'AVT055', 'AVT054', 'AVT052',
       'AVT062', 'AVT064'], dtype=object)
```

for column 2

4.DATA RESHAPING

melt()

Restructures the DataFrame from wide to long format, unpivoting the data and reshaping it based on specified identifier variables and value variables.

CODE

df.melt()

OUTPUT

```
df.melt()
```

	variable	value
0	Manufacturer	Ambarella Corp.
1	Manufacturer	Ambarella Corp.
2	Manufacturer	Ambarella Corp.
3	Manufacturer	Ambarella Corp.
4	Manufacturer	Ambarella Corp.
...	...	...
4989	Unnamed: 10	NaN
4990	Unnamed: 10	NaN
4991	Unnamed: 10	NaN
4992	Unnamed: 10	NaN
4993	Unnamed: 10	NaN

4994 rows × 2 columns

stack()

Pivots a level of the DataFrame's column labels into the innermost level of the row index, effectively converting column-wise data into row-wise data.

CODE

df.stack()

OUTPUT

```
df.stack()

0    Manufacturer                                                                  Ambarella Corp.
     Permit Number                                                                          AVT053
     DATE                                                                                3/14/2018
     VIN NUMBER                                                                  3LN6L5MU7HR609845
     VEHICLE IS CAPABLE OF OPERATING WITHOUT A DRIVER\n(Yes or No)                              No
                                                                                            ...
453  VEHICLE IS CAPABLE OF OPERATING WITHOUT A DRIVER\n(Yes or No)                              No
     DRIVER PRESENT\n(Yes or No)                                                               Yes
     DISENGAGEMENT INITIATED BY\n(AV System, Test Driver, Remote Operator, or Passenger)   Test Driver
     DISENGAGEMENT\nLOCATION\n(Interstate, Freeway, Highway, Rural Road, Street, or Parking Facility)   Downtown street
     DESCRIPTION OF FACTS CAUSING DISENGAGEMENT                  Hardware connection dropping due to a bump
Length: 4111, dtype: object
```

unstack()

Pivots a level of the DataFrame's row index into the innermost level of the column index, converting row-wise data into column-wise data.

CODE

df.unstack()

OUTPUT

```
df.unstack()
```

```
Manufacturer  0      Ambarella Corp.
              1      Ambarella Corp.
              2      Ambarella Corp.
              3      Ambarella Corp.
              4      Ambarella Corp.
                          ...
Unnamed: 10   449              NaN
              450              NaN
              451              NaN
              452              NaN
              453              NaN
Length: 4994, dtype: object
```

transpose()

Swaps the axes of the DataFrame, interchanging rows and columns, providing an alternative view of the dataset.

CODE

df.transpose()

OUTPUT

```
import pandas as pd
d=pd.read_csv("C:\\Users\\rvmut\\Desktop\\2018-19_AutonomousVehicleDisengagementReports(firsttimefilers).csv")
df=pd.DataFrame(d)
print(d)
tdf=df.transpose()
```

```
        Manufacturer Permit Number        DATE         VIN NUMBER  \
0    Ambarella Corp.        AVT053   3/14/2018  3LN6L5MU7HR609845
1    Ambarella Corp.        AVT053   3/14/2018  3LN6L5MU7HR609845
2    Ambarella Corp.        AVT053   3/14/2018  3LN6L5MU7HR609845
3    Ambarella Corp.        AVT053   3/14/2018  3LN6L5MU7HR609845
4    Ambarella Corp.        AVT053   3/15/2018  3LN6L5MU7HR609845
..               ...           ...         ...               ...
449  ThorDrive, Inc.        AVT064   4/29/2019  1FTYE1CM8JKA52066
450  ThorDrive, Inc.        AVT064  05-01-2019  1FTYE1CM8JKA52066
451  ThorDrive, Inc.        AVT064  05-06-2019  1FTYE1CM8JKA52066
452  ThorDrive, Inc.        AVT064  05-08-2019  1FTYE1CM8JKA52066
453  ThorDrive, Inc.        AVT064   6/24/2019  1FTYE1CM8JKA52066

    VEHICLE IS CAPABLE OF OPERATING WITHOUT A DRIVER\n(Yes or No)  \
0                                                   No
1                                                   No
2                                                   No
3                                                   No
4                                                   No
..                                                 ...
449                                                 No
450                                                 No
451                                                 No
452                                                 No
453                                                 No

    DRIVER PRESENT\n(Yes or No)  \
0                           Yes
```

5.DATA MERGING

Combining datasets: Merging or concatenating multiple datasets into one, useful for integrating data from different sources or files.

Joining data: Combining datasets based on common columns or indices, aligning rows with shared values for analysis or enrichment.

merge()

The merge() function in pandas combines DataFrames based on common columns or indices, aligning rows with shared values.

CODE

merge()

OUTPUT

```
df1=pd.DataFrame({'employee': ['Bob','Jake','lisa','Sue'],'group': ['Accounting','Engineering','Engineering','HR']})
df2=pd.DataFrame({'employee': ['lisa','Bob','Jake','Sue'],'hire_date': [2004,2008,2012,2014]})
display('df1','df2')
```

```
'df1'
'df2'
```

```
df3=pd.merge(df1,df2)
df3
```

	employee	group	hire_date
0	Bob	Accounting	2008
1	Jake	Engineering	2012
2	lisa	Engineering	2004
3	Sue	HR	2014

6.DATA AGGREGATION

Grouping data: Organizing dataset rows into subsets based on shared values in

one or more columns, facilitating analysis within each group.

Aggregating data: Computing summary statistics (like sum, mean, count) across grouped data, condensing information to provide insights into overall trends or patterns within the dataset.

groupby()

The groupby() function in pandas is used to split a DataFrame into groups based on one or more keys, typically corresponding to unique values in a specific column or columns.

CODE

```
df.groupby()
```

OUTPUT

```
df.groupby('Permit Number')
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x0000014A2C7A5BB0>
```

```
df.groupby('Manufacturer')
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x0000014A2C7A6270>
```

agg()

The agg() function in pandas is used to apply one or more aggregation functions (such as sum, mean, count) to the grouped data, computing summary statistics for each group.

CODE

df.agg({‘column’:’mean’})

OUTPUT

```
df.aggregate('Permit Number')
```

```
0       AVT053
1       AVT053
2       AVT053
3       AVT053
4       AVT053
         ...
449     AVT064
450     AVT064
451     AVT064
452     AVT064
453     AVT064
Name: Permit Number, Length: 454, dtype: object
```

```
df.aggregate('VIN NUMBER')
```

```
0       3LN6L5MU7HR609845
1       3LN6L5MU7HR609845
2       3LN6L5MU7HR609845
3       3LN6L5MU7HR609845
4       3LN6L5MU7HR609845
              ...
449     1FTYE1CM8JKA52066
450     1FTYE1CM8JKA52066
451     1FTYE1CM8JKA52066
452     1FTYE1CM8JKA52066
453     1FTYE1CM8JKA52066
Name: VIN NUMBER, Length: 454, dtype: object
```

7.EXPLORATORY DATA ANALYSIS

To perform EDA we need to import seaborn and matplotlib

CODE

import seaborn as sns

import matplotlib.pyplot as plt

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

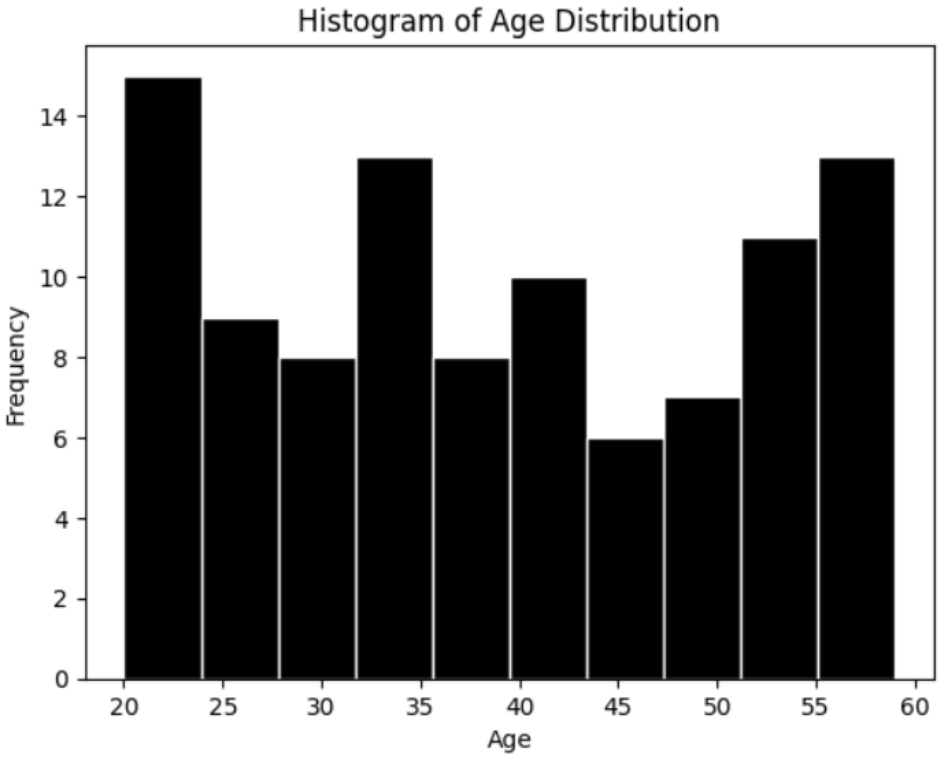
Univariate analysis: Examines a single variable's distribution and characteristics.

CODE

```
np.random.seed(0)
age_data=np.random.randint(20,60,size=100)
plt.hist(age_data,bins=10,color='black', edgecolor='white')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Histogram of Age Distribution')
plt.show()
```

OUTPUT

```
np.random.seed(0)
age_data=np.random.randint(20,60,size=100)
plt.hist(age_data,bins=10,color='black',edgecolor='white')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Histogram of Age Distribution')
plt.show()
```



Bivariate analysis: Studies the relationship between two variables.

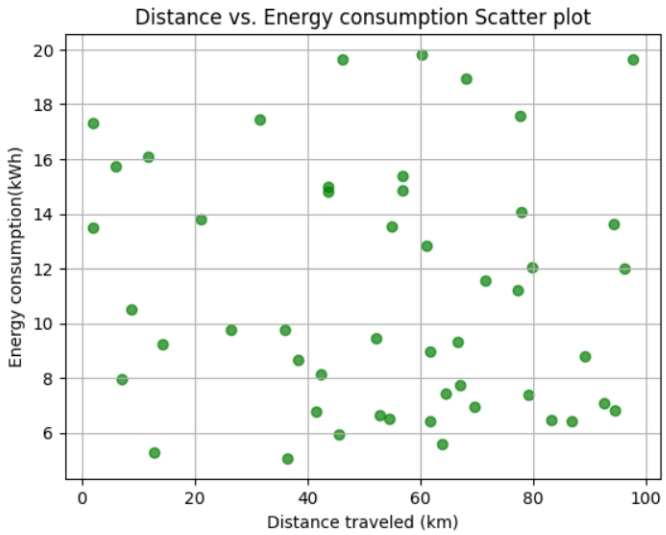
CODE

```
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(0)
```

```
distance_traveled=np.random.uniform(
0,100,size=50)

energy_consumption=np.random.unifor
m(5,20,size=50)

plt.scatter(distance_traveled,energy_co
nsumption,color='green',alpha=0.7)

plt.xlabel('Distance traveled
(km)'),plt.ylabel('Energy
consumption(kWh)'),plt.title('Distance
vs. Energy consumption Scatter plot')

plt.grid(True)

plt.show()
```

OUTPUT

```
import matplotlib.pyplot as plt
np.random.seed(0)
distance_traveled=np.random.uniform(0,100,size=50)
energy_consumption=np.random.uniform(5,20,size=50)
plt.scatter(distance_traveled,energy_consumption,color='green',alpha=0.7)
plt.xlabel('Distance traveled (km)'),plt.ylabel('Energy consumption(kWh)'),plt.title('Distance vs. Energy consumption Scatter plot')
plt.grid(True)
plt.show()
```



Multivariate analysis: Analyzes relationships among three or more variables simultaneously.

CODE

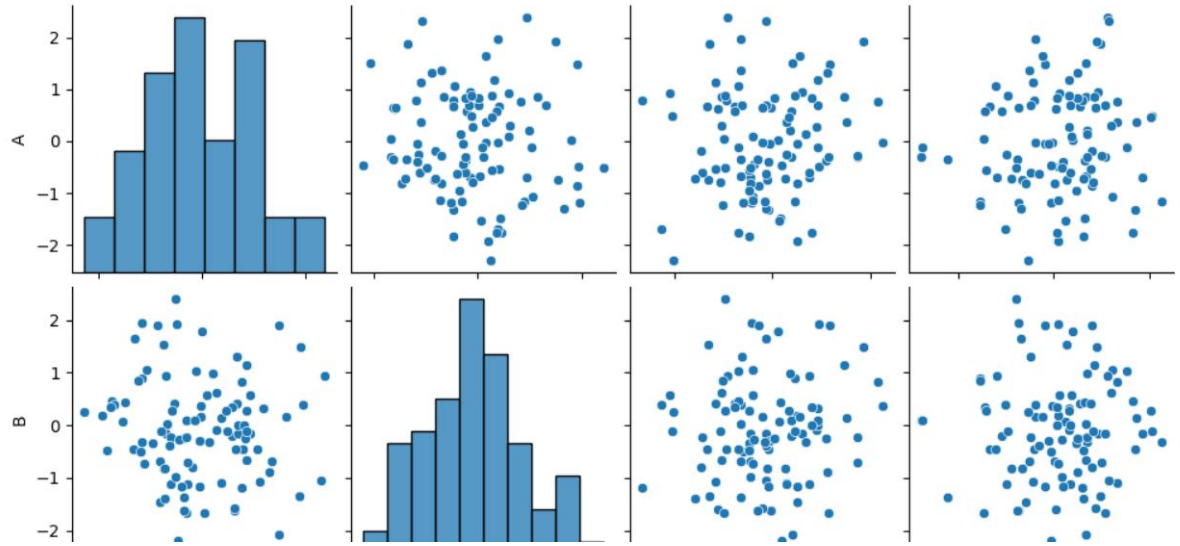
```
import seaborn as sns
import pandas as pd
import numpy as np
data=np.random.randn(100,4)
df=pd.DataFrame(data,columns=['A','B','C','D'])
```

sns.pairplot(df)

OUTPUT

```
import seaborn as sns
import pandas as pd
import numpy as np
data=np.random.randn(100,4)
df=pd.DataFrame(data,columns=['A','B','C','D'])
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x2648618a570>
```



8.FEATURE ENGINEERING

Creating user profiles: Constructing individual profiles based on specific user characteristics or behaviors, often derived from data such as demographics, preferences, and interactions.

CODE

features[f'{sensor}_mean'] = np.mean(data)

features[f'{sensor}_std'] = np.std(data)

features[f'{sensor}_max'] = np.max(data)

features[f'{sensor}_min'] = np.min(data)

OUTPUT

```
import numpy as np
sensor_data = {
    'lidar': np.random.rand(100),
    'camera': np.random.rand(100),
    'radar': np.random.rand(100)
}
features = {}
for sensor, data in sensor_data.items():
    features[f'{sensor}_mean'] = np.mean(data)
    features[f'{sensor}_std'] = np.std(data)
    features[f'{sensor}_max'] = np.max(data)
    features[f'{sensor}_min'] = np.min(data)

features['lidar_camera_corr'] = np.corrcoef(sensor_data['lidar'], sensor_data['camera'])[0, 1]
features['radar_camera_corr'] = np.corrcoef(sensor_data['radar'], sensor_data['camera'])[0, 1]
features['lidar_radar_corr'] = np.corrcoef(sensor_data['lidar'], sensor_data['radar'])[0, 1]

print(features)
```

```
{'lidar_mean': 0.5386727421385225, 'lidar_std': 0.2924226291719068, 'lidar_max': 0.9943416834558875, 'lidar_min': 0.006599144246275168, 'camera_mean': 0.52
44283634348283, 'camera_std': 0.2683279481687l425, 'camera_max': 0.9960252280767453, 'camera_min': 0.016454824426866854, 'radar_mean': 0.529043781918893, '
radar_std': 0.2968127111255024, 'radar_max': 0.9952452918222159, 'radar_min': 0.010877807857149757, 'lidar_camera_corr': -0.06573438956731162, 'radar_came
ra_corr': -0.027918955983407788, 'lidar_radar_corr': 0.14742159916241032}
```

Temporal analysis: Examining data trends, patterns, or changes over time,

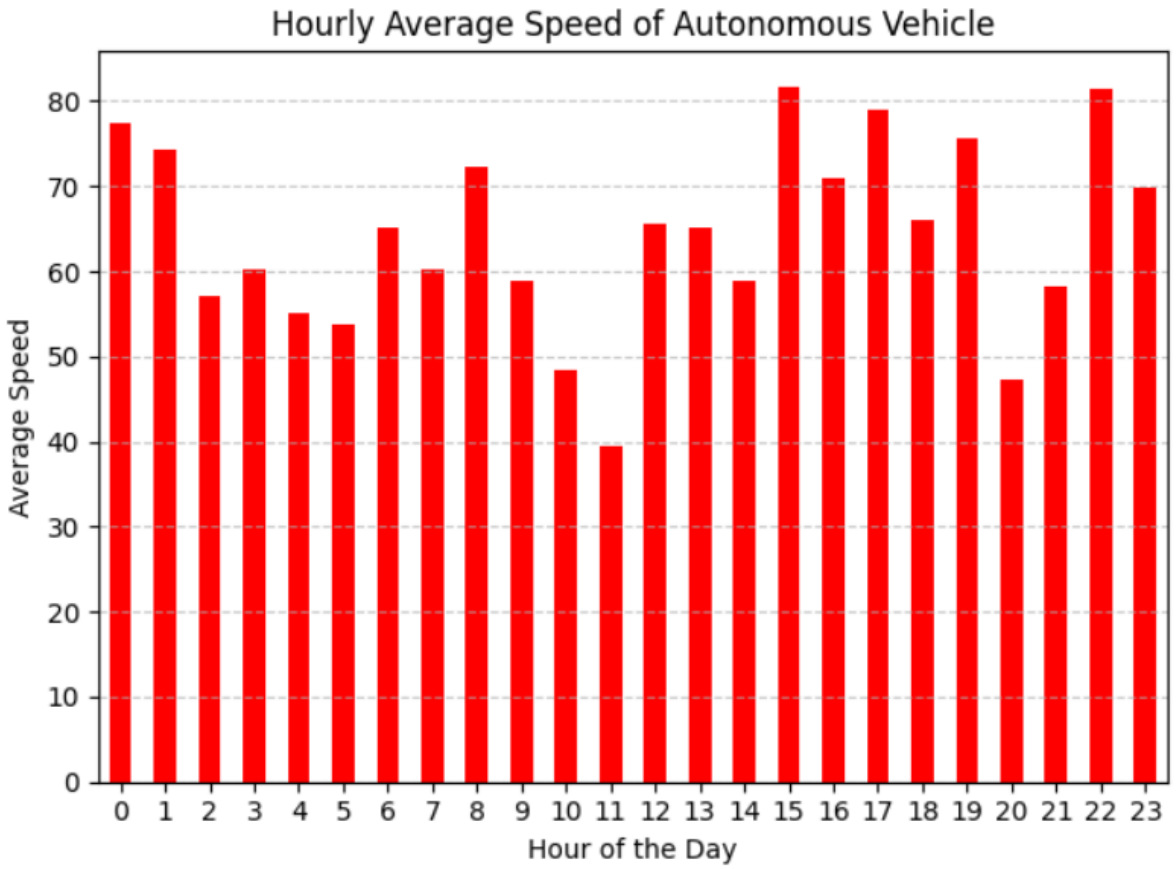
allowing for insights into temporal variations and developments.

CODE

```
timestamps = pd.date_range('2024-05-01', periods=100, freq='h')

speed_data = np.random.randint(30, 100, size=len(timestamps))
```

OUTPUT

```
plt.xlabel('Hour of the Day')
plt.ylabel('Average Speed')
plt.xticks(rotation=0)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



Content embeddings: Content embeddings of 2019 autonomous disengagement reports: Transforming textual content from the 2019 autonomous disengagement reports into

dense vector representations, enabling computational analysis and comparison based on semantic similarities between different reports.

CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import nltk
nltk.download('punkt')
corpus = [
    "Autonomous vehicles are the future of transportation",
    "Self-driving cars will revolutionize commuting",
    "Artificial intelligence powers autonomous vehicles",
```

```
    "Driverless cars use advanced
sensors and algorithms",
    "The adoption of autonomous
vehicles is increasing rapidly"
]
tokenized_corpus =
[nltk.word_tokenize(sentence.lower())
for sentence in corpus]
word_to_index = {word: idx for idx,
word in enumerate(set(word for
sentence in tokenized_corpus for word
in sentence))}
index_to_word = {idx: word for word,
idx in word_to_index.items()}
vocab_size = len(word_to_index)
window_size = 2
data = []
for sentence in tokenized_corpus:
```

```
    for i, word in enumerate(sentence):
        for j in range(max(i - window_size, 0), min(i + window_size + 1, len(sentence))):
            if j != i:

data.append((word_to_index[word], word_to_index[sentence[j]]))
df = pd.DataFrame(data, columns=['input', 'output'])
input_dim = vocab_size
hidden_dim = 100
output_dim = vocab_size
np.random.seed(42)
W1 = np.random.randn(input_dim, hidden_dim)
```

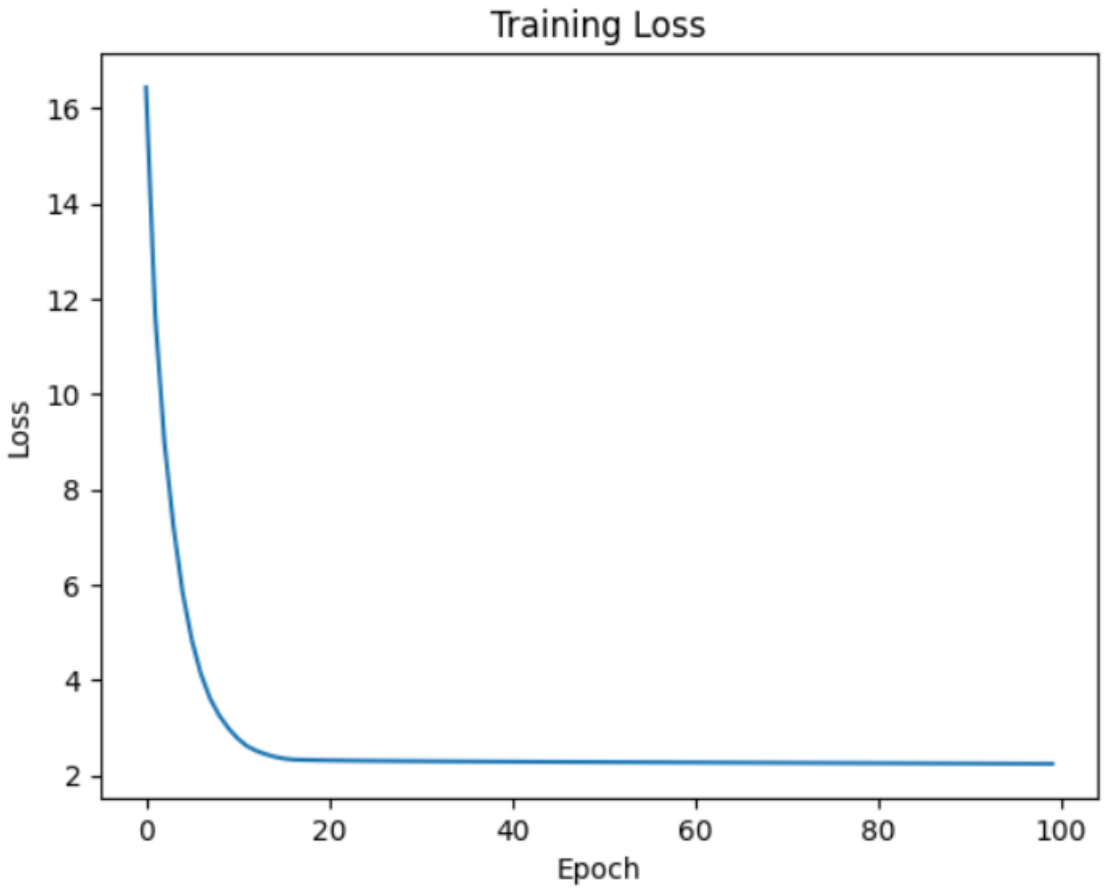
```
W2 = np.random.randn(hidden_dim, output_dim)
learning_rate = 0.01
epochs = 100
losses = []
for epoch in range(epochs):
    epoch_loss = 0
    for _, row in df.iterrows():
        x = np.zeros(input_dim)
        x[row['input']] = 1
        y_true = np.zeros(output_dim)
        y_true[row['output']] = 1
        hidden_layer = np.dot(x, W1)
        output_layer = np.dot(hidden_layer, W2)
                exp_scores = np.exp(output_layer)
```

```
        probs = exp_scores /
np.sum(exp_scores)
                loss = -
np.log(probs[np.argmax(y_true)])
        epoch_loss += loss
                delta_output = probs - y_true
        dW2 = np.outer(hidden_layer,
delta_output)
        delta_hidden =
np.dot(delta_output, W2.T)
        dW1 = np.outer(x, delta_hidden)
                W1 -= learning_rate * dW1
        W2 -= learning_rate * dW2
      losses.append(epoch_loss / len(df))
    if (epoch+1) % 10 == 0:
        print(f'Epoch {epoch+1}, Loss:
{epoch_loss / len(df):.4f}')
```

```
plt.plot(range(epochs), losses)
plt.title('Training Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.show()

word_embeddings = W1
```

OUTPUT

```
Epoch 20, Loss: 2.3182
Epoch 30, Loss: 2.2987
Epoch 40, Loss: 2.2879
Epoch 50, Loss: 2.2792
Epoch 60, Loss: 2.2713
Epoch 70, Loss: 2.2640
Epoch 80, Loss: 2.2569
Epoch 90, Loss: 2.2501
Epoch 100, Loss: 2.2434
```



ASSUMED SCENARIO

In 2019, the field of autonomous vehicles was in a dynamic phase, marked by the release of comprehensive disengagement reports. These reports meticulously documented

scenarios where autonomous vehicles encountered challenges that necessitated human intervention. Such challenges ranged from technical glitches to navigating through complex and unpredictable driving conditions.

Each disengagement report served as a window into the intricate world of autonomous driving, offering insights into the myriad scenarios these vehicles encountered on the road. From sudden road closures to unanticipated obstacles, the reports meticulously cataloged the circumstances leading to disengagements.

However, amidst these challenges lay the underlying objectives of these

interventions, which often revolved around ensuring the safety of passengers, pedestrians, and other vehicles on the road. By allowing human operators to take control when necessary, autonomous systems aimed to mitigate potential risks and ensure smooth navigation through diverse environments.

Moreover, these reports didn't merely focus on the setbacks encountered by autonomous vehicles; they also highlighted positive outcomes gleaned from the dataset analysis. These positive outcomes underscored the advancements made by autonomous systems, showcasing instances where they successfully alerted human

operators in time to avert potential hazards or navigated complex scenarios with precision.

Overall, the 2019 autonomous disengagement reports played a pivotal role in fostering transparency within the autonomous driving industry. They provided stakeholders with valuable insights into the performance of autonomous systems, shedding light on both the challenges faced and the strides made toward safer and more reliable autonomous transportation.

CONCLUSION

The analysis of the 2019 autonomous disengagement reports involves various techniques and

methodologies, including data preprocessing (such as handling missing values and transposing datasets), data validation (using functions like isnull().sum()), and exploratory data analysis (like univariate, bivariate, and multivariate analysis). Additionally, feature engineering and temporal analysis offer deeper insights into the performance and trends of autonomous driving technology over time. Furthermore, there's potential for more advanced analyses such as creating user profiles and content embeddings to extract valuable information from the reports. Overall, a comprehensive examination of the 2019 autonomous disengagement reports using these methods can provide valuable insights into the state and

evolution of autonomous vehicle technology during that period.