INSY 5339 - 002

Principles of Business Data Mining

Project proposal - Urban Traffic Densities in Cities

Group-6

Abhishek Babu Madini - ID: 1002177917
Nikhita Kalburgikar – ID: 1002160264
Nivetha Thangaraj – ID: 1002154961
Vaishnavi Palle – ID: 1002172756
Yuvarekha Mahendran - ID:1002175347

Source:

https://www.kaggle.com/datasets/tanishqdublish/urban-traffic-density-in-cities

Project description:

The dataset gives the basic understanding of traffic conditions in specific cities by considering specific key factors like vehicle type, weather conditions, economic conditions, day of week, hour of day, speed, energy consumption, peak hour, and random event.

How many observations in the dataset?	1048576
How many binary/categorical variables?	There are 2 binary variables (is peak hour, random event occurred), There are 5 categorical variables (city, vehicle type, weather conditions, economic conditions, day of week)
How many continuous variables?	There are 6 continuous variables like hour of day, speed, is peak hour, random event occurred, energy consumption, traffic density)
What is the outcome / target variable?	Traffic density
If binary or categorical: What percentage of the variables belong to each class	Binary variables – • Is Peak Hour 0 84.530083 1 15.469917 Random Event Occurred 0 95.011262 1 4.988738

<table>
<tr><td></td><td>Categorical variables –

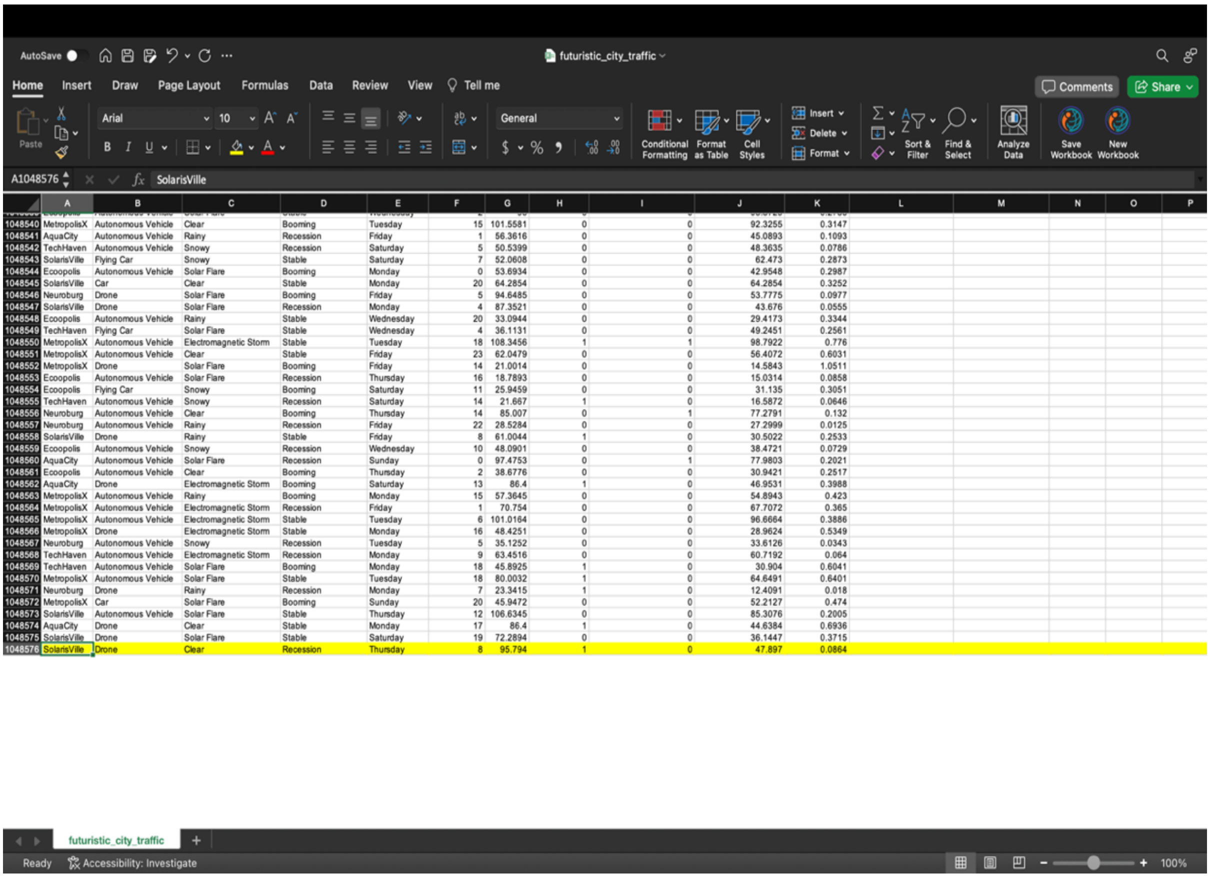
• City
Ecoopolis 16.741926
AquaCity 16.678460
Neuroburg 16.660339
SolarisVille 16.643694
MetropolisX 16.643694
TechHaven 16.631887

Vehicle Type
Autonomous Vehicle 62.108437
Drone 25.004858
Flying Car 6.466557
Car 6.420147
Weather
Solar Flare 20.026534
Snowy 20.023090
Electromagnetic Storm 19.995375
Clear 19.978402
Rainy 19.976598

Economic Condition
Booming 33.346589
Recession 33.337324
Stable 33.316087

Day Of Week
Tuesday 14.331562
Wednesday 14.331152
Thursday 14.309095
Saturday 14.275312
Monday 14.259569
Sunday 14.252845
Friday 14.240464</td></tr>
<tr><td>If continuous: What is the mean value of the target variable?</td><td>0.2771</td></tr>
<tr><td>Before doing any further processing, what would your prediction of the target variable be?</td><td>Traffic density is our target variable which is continuous</td></tr>
</table>

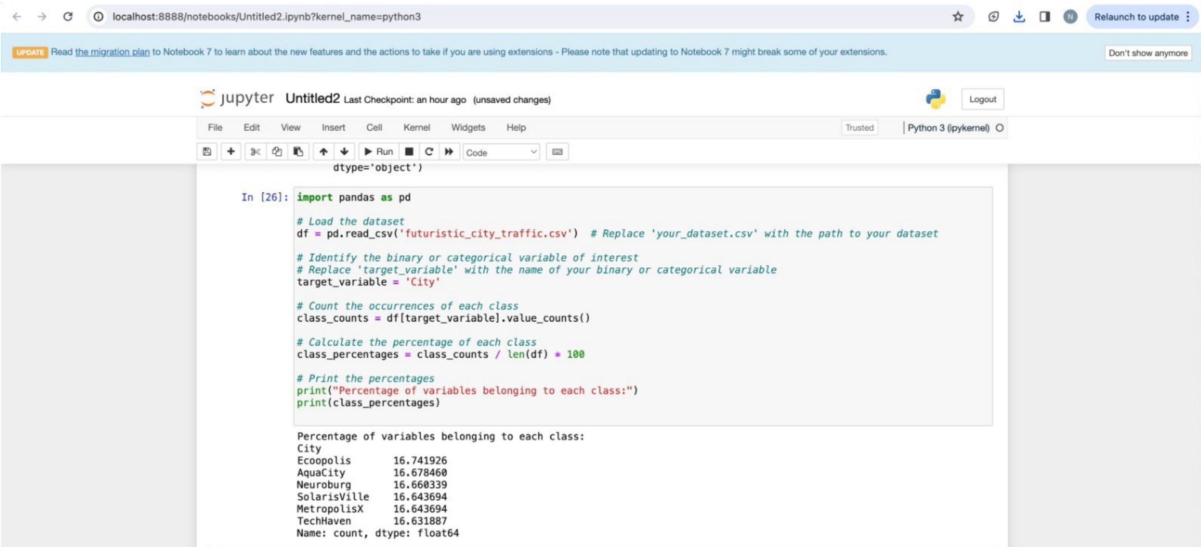
How many observations in the dataset?



	A	B	C	D	E	F	G	H	I	J	K
1048540	MetropolisX	Autonomous Vehicle	Clear	Booming	Tuesday	15	101.5581	0	0	92.3255	0.3147
1048541	AquaCity	Autonomous Vehicle	Rainy	Recession	Friday	1	56.3616	0	0	45.0893	0.1093
1048542	TechHaven	Autonomous Vehicle	Snowy	Recession	Saturday	5	50.5399	0	0	48.3635	0.0786
1048543	SolarisVille	Flying Car	Snowy	Stable	Saturday	7	52.0608	0	0	62.473	0.2873
1048544	Ecoopolis	Autonomous Vehicle	Solar Flare	Booming	Monday	0	53.6934	0	0	42.9548	0.2987
1048545	SolarisVille	Car	Clear	Stable	Monday	20	64.2854	0	0	64.2854	0.3252
1048546	Neuroburg	Drone	Solar Flare	Booming	Friday	5	94.6485	0	0	53.7775	0.0977
1048547	SolarisVille	Drone	Solar Flare	Recession	Monday	4	87.3521	0	0	43.676	0.0555
1048548	Ecoopolis	Autonomous Vehicle	Rainy	Stable	Wednesday	20	33.0944	0	0	29.4173	0.3344
1048549	TechHaven	Flying Car	Solar Flare	Stable	Wednesday	4	36.1131	0	0	49.2451	0.2561
1048550	MetropolisX	Autonomous Vehicle	Electromagnetic Storm	Stable	Tuesday	18	108.3456	1	1	98.7922	0.776
1048551	MetropolisX	Autonomous Vehicle	Clear	Stable	Friday	23	62.0479	0	0	56.4072	0.6031
1048552	MetropolisX	Drone	Solar Flare	Booming	Friday	14	21.0014	0	0	14.5843	1.0511
1048553	Ecoopolis	Autonomous Vehicle	Solar Flare	Recession	Thursday	16	18.7893	0	0	15.0314	0.0858
1048554	Ecoopolis	Flying Car	Snowy	Booming	Saturday	11	25.9459	0	0	31.135	0.3051
1048555	TechHaven	Autonomous Vehicle	Snowy	Recession	Saturday	14	21.667	1	0	16.5872	0.0646
1048556	Neuroburg	Autonomous Vehicle	Clear	Booming	Thursday	14	85.007	0	1	77.2791	0.132
1048557	Neuroburg	Autonomous Vehicle	Rainy	Recession	Friday	22	28.5284	0	0	27.2999	0.0125
1048558	SolarisVille	Drone	Rainy	Stable	Friday	8	61.0044	1	0	30.5022	0.2533
1048559	Ecoopolis	Autonomous Vehicle	Snowy	Recession	Wednesday	10	48.0901	0	0	38.4721	0.0729
1048560	AquaCity	Autonomous Vehicle	Solar Flare	Recession	Sunday	0	97.4753	0	1	77.9803	0.2021
1048561	Ecoopolis	Autonomous Vehicle	Clear	Booming	Thursday	2	38.6776	0	0	30.9421	0.2517
1048562	AquaCity	Drone	Electromagnetic Storm	Booming	Saturday	13	86.4	1	0	46.9531	0.3988
1048563	MetropolisX	Autonomous Vehicle	Rainy	Booming	Monday	15	57.3645	0	0	54.8943	0.423
1048564	MetropolisX	Autonomous Vehicle	Electromagnetic Storm	Recession	Friday	1	70.754	0	0	67.7072	0.365
1048565	MetropolisX	Autonomous Vehicle	Electromagnetic Storm	Stable	Tuesday	6	101.0164	0	0	96.6664	0.3886
1048566	MetropolisX	Drone	Electromagnetic Storm	Stable	Monday	16	48.4251	0	0	28.9624	0.5349
1048567	Neuroburg	Autonomous Vehicle	Snowy	Recession	Tuesday	5	35.1252	0	0	33.6126	0.0343
1048568	TechHaven	Autonomous Vehicle	Electromagnetic Storm	Recession	Monday	9	63.4516	0	0	60.7192	0.064
1048569	TechHaven	Autonomous Vehicle	Solar Flare	Booming	Monday	18	45.8925	1	0	30.904	0.6041
1048570	MetropolisX	Autonomous Vehicle	Solar Flare	Stable	Tuesday	18	80.0032	1	0	64.6491	0.6401
1048571	Neuroburg	Drone	Rainy	Recession	Monday	7	23.3415	1	0	12.4091	0.018
1048572	MetropolisX	Car	Solar Flare	Booming	Sunday	20	45.9472	0	0	52.2127	0.474
1048573	SolarisVille	Autonomous Vehicle	Solar Flare	Stable	Thursday	12	106.6345	0	0	85.3076	0.2005
1048574	AquaCity	Drone	Clear	Stable	Monday	17	86.4	1	0	44.6384	0.6936
1048575	SolarisVille	Drone	Solar Flare	Stable	Saturday	19	72.2894	0	0	36.1447	0.3715
1048576	SolarisVille	Drone	Clear	Recession	Thursday	8	95.794	1	0	47.897	0.0864

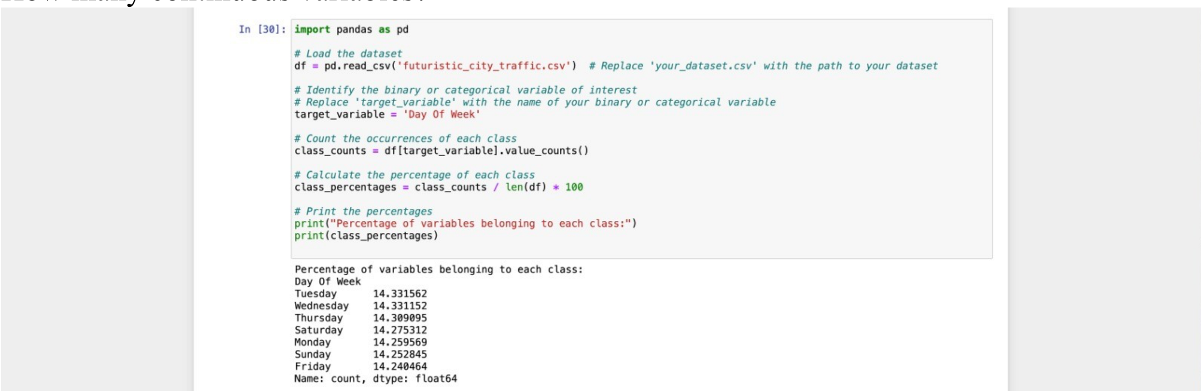
How many binary/categorical variables?

The binary variables are Is peak hour and random event occurred.



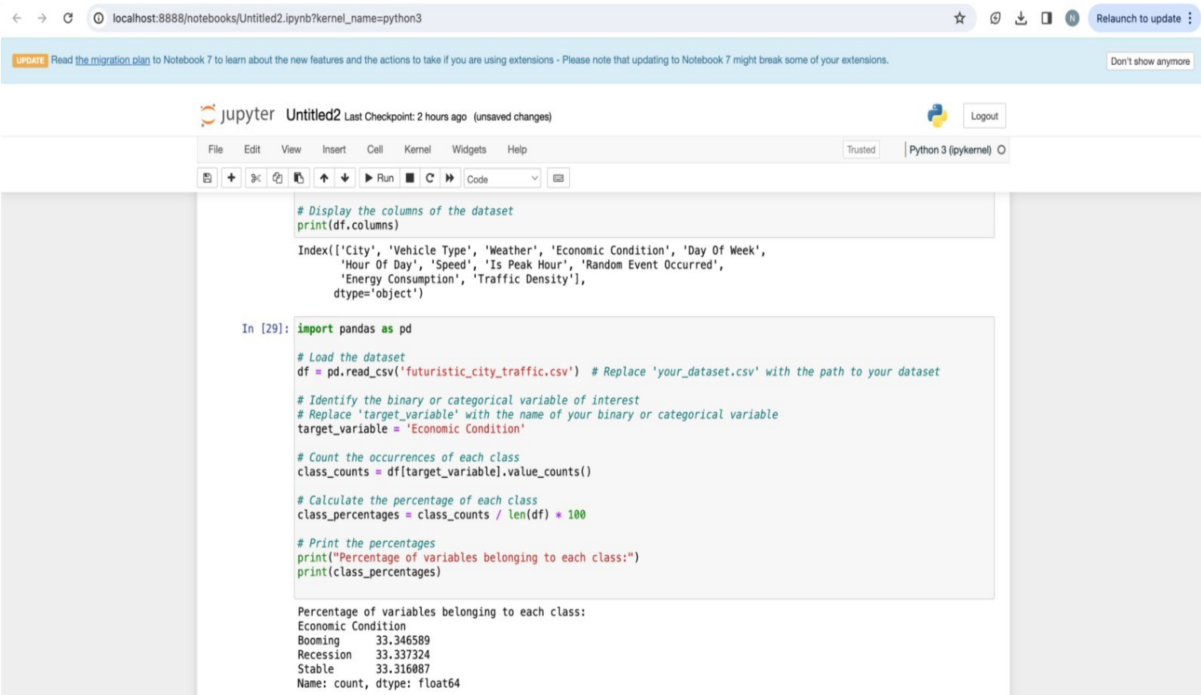

```
      dtype='object')

In [26]: import pandas as pd

# Load the dataset
df = pd.read_csv('futuristic_city_traffic.csv')  # Replace 'your_dataset.csv' with the path to your dataset

# Identify the binary or categorical variable of interest
# Replace 'target_variable' with the name of your binary or categorical variable
target_variable = 'City'

# Count the occurrences of each class
class_counts = df[target_variable].value_counts()

# Calculate the percentage of each class
class_percentages = class_counts / len(df) * 100

# Print the percentages
print("Percentage of variables belonging to each class:")
print(class_percentages)

Percentage of variables belonging to each class:
City
Ecoopolis       16.741926
AquaCity        16.678460
Neuroburg       16.660339
SolarisVille    16.643694
MetropolisX     16.643694
TechHaven       16.631887
Name: count, dtype: float64
```

How many continuous variables?



```
In [30]: import pandas as pd

# Load the dataset
df = pd.read_csv('futuristic_city_traffic.csv')  # Replace 'your_dataset.csv' with the path to your dataset

# Identify the binary or categorical variable of interest
# Replace 'target_variable' with the name of your binary or categorical variable
target_variable = 'Day Of Week'

# Count the occurrences of each class
class_counts = df[target_variable].value_counts()

# Calculate the percentage of each class
class_percentages = class_counts / len(df) * 100

# Print the percentages
print("Percentage of variables belonging to each class:")
print(class_percentages)

Percentage of variables belonging to each class:
Day Of Week
Tuesday      14.331562
Wednesday    14.331152
Thursday     14.309095
Saturday     14.275312
Monday       14.259569
Sunday       14.252845
Friday       14.240464
Name: count, dtype: float64
```

If binary or categorical: What percentage of the variables belong to each class.




```
# Display the columns of the dataset
print(df.columns)

Index(['City', 'Vehicle Type', 'Weather', 'Economic Condition', 'Day Of Week',
       'Hour Of Day', 'Speed', 'Is Peak Hour', 'Random Event Occurred',
       'Energy Consumption', 'Traffic Density'],
      dtype='object')

In [29]: import pandas as pd

# Load the dataset
df = pd.read_csv('futuristic_city_traffic.csv')  # Replace 'your_dataset.csv' with the path to your dataset

# Identify the binary or categorical variable of interest
# Replace 'target_variable' with the name of your binary or categorical variable
target_variable = 'Economic Condition'

# Count the occurrences of each class
class_counts = df[target_variable].value_counts()

# Calculate the percentage of each class
class_percentages = class_counts / len(df) * 100

# Print the percentages
print("Percentage of variables belonging to each class:")
print(class_percentages)

Percentage of variables belonging to each class:
Economic Condition
Booming      33.346589
Recession    33.337324
Stable       33.316087
Name: count, dtype: float64
```

localhost:8888/notebooks/Untitled2.ipynb?kernel_name=python3

Relaunch to update

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

Don't show anymore

jupyter Untitled2 Last Checkpoint: an hour ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

```
      dtype='object')
```

In [26]:

```python
import pandas as pd

# Load the dataset
df = pd.read_csv('futuristic_city_traffic.csv')  # Replace 'your_dataset.csv' with the path to your dataset

# Identify the binary or categorical variable of interest
# Replace 'target_variable' with the name of your binary or categorical variable
target_variable = 'City'

# Count the occurrences of each class
class_counts = df[target_variable].value_counts()

# Calculate the percentage of each class
class_percentages = class_counts / len(df) * 100

# Print the percentages
print("Percentage of variables belonging to each class:")
print(class_percentages)
```

```
Percentage of variables belonging to each class:
City
Ecoopolis       16.741926
AquaCity        16.678460
Neuroburg       16.660339
SolarisVille    16.643694
MetropolisX     16.643694
TechHaven       16.631887
Name: count, dtype: float64
```


localhost:8888/notebooks/Untitled2.ipynb?kernel_name=python3

Relaunch to update

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

Don't show anymore

jupyter Untitled2 Last Checkpoint: an hour ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

```
      dtype='object')
```

In [26]:

```python
import pandas as pd

# Load the dataset
df = pd.read_csv('futuristic_city_traffic.csv')  # Replace 'your_dataset.csv' with the path to your dataset

# Identify the binary or categorical variable of interest
# Replace 'target_variable' with the name of your binary or categorical variable
target_variable = 'City'

# Count the occurrences of each class
class_counts = df[target_variable].value_counts()

# Calculate the percentage of each class
class_percentages = class_counts / len(df) * 100

# Print the percentages
print("Percentage of variables belonging to each class:")
print(class_percentages)
```

```
Percentage of variables belonging to each class:
City
Ecoopolis       16.741926
AquaCity        16.678460
Neuroburg       16.660339
SolarisVille    16.643694
MetropolisX     16.643694
TechHaven       16.631887
Name: count, dtype: float64
```

localhost:8888/notebooks/Untitled2.ipynb?kernel_name=python3

Relaunch to update

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions. Don't show anymore

jupyter Untitled2 Last Checkpoint: 2 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
# Display the columns of the dataset
print(df.columns)
```

```
Index(['City', 'Vehicle Type', 'Weather', 'Economic Condition', 'Day Of Week',
       'Hour Of Day', 'Speed', 'Is Peak Hour', 'Random Event Occurred',
       'Energy Consumption', 'Traffic Density'],
      dtype='object')
```

In [29]:

```
import pandas as pd

# Load the dataset
df = pd.read_csv('futuristic_city_traffic.csv')  # Replace 'your_dataset.csv' with the path to your dataset

# Identify the binary or categorical variable of interest
# Replace 'target_variable' with the name of your binary or categorical variable
target_variable = 'Economic Condition'

# Count the occurrences of each class
class_counts = df[target_variable].value_counts()

# Calculate the percentage of each class
class_percentages = class_counts / len(df) * 100

# Print the percentages
print("Percentage of variables belonging to each class:")
print(class_percentages)
```

```
Percentage of variables belonging to each class:
Economic Condition
Booming      33.346589
Recession    33.337324
Stable       33.316087
Name: count, dtype: float64
```

In [30]:

```
import pandas as pd

# Load the dataset
df = pd.read_csv('futuristic_city_traffic.csv')  # Replace 'your_dataset.csv' with the path to your dataset

# Identify the binary or categorical variable of interest
# Replace 'target_variable' with the name of your binary or categorical variable
target_variable = 'Day Of Week'

# Count the occurrences of each class
class_counts = df[target_variable].value_counts()

# Calculate the percentage of each class
class_percentages = class_counts / len(df) * 100

# Print the percentages
print("Percentage of variables belonging to each class:")
print(class_percentages)
```

```
Percentage of variables belonging to each class:
Day Of Week
Tuesday      14.331562
Wednesday    14.331152
Thursday     14.309095
Saturday     14.275312
Monday       14.259569
Sunday       14.252845
Friday       14.240464
Name: count, dtype: float64
```


localhost:8888/notebooks/Untitled2.ipynb?kernel_name=python3

Relaunch to update

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions. Don't show anymore

jupyter Untitled2 Last Checkpoint: 2 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
# Display the columns of the dataset
print(df.columns)
```

```
Index(['City', 'Vehicle Type', 'Weather', 'Economic Condition', 'Day Of Week',
       'Hour Of Day', 'Speed', 'Is Peak Hour', 'Random Event Occurred',
       'Energy Consumption', 'Traffic Density'],
      dtype='object')
```

In [29]:

```
import pandas as pd

# Load the dataset
df = pd.read_csv('futuristic_city_traffic.csv')  # Replace 'your_dataset.csv' with the path to your dataset

# Identify the binary or categorical variable of interest
# Replace 'target_variable' with the name of your binary or categorical variable
target_variable = 'Economic Condition'

# Count the occurrences of each class
class_counts = df[target_variable].value_counts()

# Calculate the percentage of each class
class_percentages = class_counts / len(df) * 100

# Print the percentages
print("Percentage of variables belonging to each class:")
print(class_percentages)
```

```
Percentage of variables belonging to each class:
Economic Condition
Booming      33.346589
Recession    33.337324
Stable       33.316087
Name: count, dtype: float64
```

```
In [30]: import pandas as pd

         # Load the dataset
         df = pd.read_csv('futuristic_city_traffic.csv')  # Replace 'your_dataset.csv' with the path to your dataset

         # Identify the binary or categorical variable of interest
         # Replace 'target_variable' with the name of your binary or categorical variable
         target_variable = 'Day Of Week'

         # Count the occurrences of each class
         class_counts = df[target_variable].value_counts()

         # Calculate the percentage of each class
         class_percentages = class_counts / len(df) * 100

         # Print the percentages
         print("Percentage of variables belonging to each class:")
         print(class_percentages)

         Percentage of variables belonging to each class:
         Day Of Week
         Tuesday      14.331562
         Wednesday    14.331152
         Thursday     14.309095
         Saturday     14.275312
         Monday       14.259569
         Sunday       14.252845
         Friday       14.240464
         Name: count, dtype: float64
```

If continuous: What is the mean value of the target variable?

futuristic_city_traffic

M20

	A	B	C	D	E	F	G	H	I	J	K	L
1	City	Vehicle Type	Weather	Economic Condition	Day Of Week	Hour Of Day	Speed	Is Peak Hour	Random Event Occurred	Energy Consumption	Traffic Density	
2	SolarisVille	Drone	Snowy	Stable	Sunday	20	29.4268	0	0	14.7134	0.5241	
3	AquaCity	Flying Car	Solar Flare	Recession	Wednesday	2	118.8	0	0	143.5682	0.3208	
4	Neuroburg	Autonomous Vehicle	Solar Flare	Recession	Wednesday	16	100.3904	0	0	91.264	0.0415	
5	Ecoopolis	Drone	Clear	Booming	Thursday	8	76.8	1	0	46.0753	0.1811	
6	AquaCity	Autonomous Vehicle	Solar Flare	Stable	Saturday	16	45.2176	0	0	40.1934	0.4544	
7	TechHaven	Autonomous Vehicle	Snowy	Recession	Thursday	20	30.5179	0	0	37.5562	0.0843	
8	Ecoopolis	Autonomous Vehicle	Rainy	Recession	Monday	21	43.9222	0	0	39.042	0.0293	
9	Ecoopolis	Autonomous Vehicle	Snowy	Stable	Friday	20	20.4236	0	0	18.1543	0.1393	
10	MetropolisX	Drone	Snowy	Recession	Saturday	7	69.9735	0	0	41.8502	0.1505	mean of traffic density
11	SolarisVille	Drone	Snowy	Booming	Saturday	15	37.9431	0	0	18.9715	0.2485	0.277125007
12	Neuroburg	Drone	Snowy	Stable	Saturday	13	29.5809	1	0	14.1535	0.0626	
13	MetropolisX	Drone	Clear	Booming	Tuesday	7	40.4778	1	0	18.399	0.5179	
14	TechHaven	Autonomous Vehicle	Clear	Recession	Sunday	7	87.8801	0	0	79.891	0.1248	
15	Neuroburg	Autonomous Vehicle	Electromagnetic Storm	Recession	Sunday	10	60.0447	0	0	57.459	0.0342	
16	Ecoopolis	Drone	Solar Flare	Booming	Monday	8	70.2698	1	0	35.1349	0.2389	
17	Ecoopolis	Car	Rainy	Booming	Friday	9	66.8856	0	0	66.8856	0.2521	
18	Neuroburg	Autonomous Vehicle	Snowy	Booming	Monday	21	126.7644	0	0	107.1712	0.0784	
19	MetropolisX	Flying Car	Clear	Recession	Friday	21	87.0448	0	0	118.6975	0.1352	
20	Ecoopolis	Drone	Solar Flare	Booming	Friday	3	62.3252	0	0	31.1626	0.2185	
21	TechHaven	Autonomous Vehicle	Solar Flare	Stable	Wednesday	19	67.2491	0	0	56.607	0.1994	
22	Ecoopolis	Autonomous Vehicle	Rainy	Stable	Saturday	1	69.0747	0	0	55.2598	0.2113	
23	Ecoopolis	Autonomous Vehicle	Electromagnetic Storm	Recession	Saturday	19	32.5679	0	0	26.0543	0.055	
24	Ecoopolis	Car	Rainy	Recession	Thursday	5	94.7111	0	0	94.7111	0.0984	
25	SolarisVille	Autonomous Vehicle	Snowy	Recession	Monday	10	118.7922	0	0	95.0338	0.0588	
26	MetropolisX	Flying Car	Clear	Booming	Friday	12	114.4823	0	0	156.1122	0.4281	
27	Neuroburg	Drone	Electromagnetic Storm	Recession	Thursday	23	89.2206	0	0	53.3616	0.0184	
28	Ecoopolis	Autonomous Vehicle	Solar Flare	Recession	Saturday	10	42.6291	0	0	34.1033	0.134	
29	MetropolisX	Autonomous Vehicle	Clear	Recession	Wednesday	12	37.6441	0	0	34.2219	0.2449	
30	Ecoopolis	Drone	Clear	Recession	Monday	13	93.2075	0	0	46.6037	0.0446	
31	Ecoopolis	Autonomous Vehicle	Clear	Stable	Thursday	0	96	0	0	90.1531	0.2053	
32	Ecoopolis	Autonomous Vehicle	Rainy	Recession	Sunday	10	45.5341	0	0	36.4273	0.1085	
33	Neuroburg	Autonomous Vehicle	Solar Flare	Recession	Wednesday	1	69.4651	0	0	63.1501	0.0219	
34	Ecoopolis	Autonomous Vehicle	Snowy	Booming	Thursday	2	22.7334	0	0	22.7334	0.1498	
35	Ecoopolis	Autonomous Vehicle	Snowy	Booming	Tuesday	13	55.0833	0	0	44.0666	0.157	
36	Ecoopolis	Autonomous Vehicle	Snowy	Stable	Tuesday	11	27.5094	0	1	22.0075	0.1949	
37	MetropolisX	Autonomous Vehicle	Electromagnetic Storm	Stable	Thursday	20	70.3384	0	0	107.9196	0.6792	
38	TechHaven	Autonomous Vehicle	Snowy	Stable	Friday	21	55.6397	0	0	49.2998	0.1642	
39	AquaCity	Flying Car	Rainy	Booming	Monday	17	39.386	1	0	47.2632	0.5913	
40	TechHaven	Drone	Clear	Stable	Thursday	17	45.6719	1	0	20.76	0.2946	
41	TechHaven	Autonomous Vehicle	Electromagnetic Storm	Booming	Wednesday	12	24.7208	0	0	23.6562	0.3319	
42	Neuroburg	Autonomous Vehicle	Clear	Booming	Saturday	4	44.616	0	0	40.56	0.0803	
43	TechHaven	Autonomous Vehicle	Clear	Recession	Thursday	23	57.7985	0	0	52.5441	0.0457	
44	Ecoopolis	Autonomous Vehicle	Clear	Booming	Saturday	11	24.8612	0	0	19.8889	0.223	
45	TechHaven	Autonomous Vehicle	Solar Flare	Booming	Tuesday	12	87.8502	0	0	79.8638	0.2282	
46	Ecoopolis	Autonomous Vehicle	Snowy	Stable	Saturday	18	58.8387	0	0	47.071	0.2684	
47	MetropolisX	Autonomous Vehicle	Snowy	Stable	Thursday	17	53.0764	1	0	40.6326	0.5668	
48	SolarisVille	Autonomous Vehicle	Rainy	Recession	Thursday	15	30.3887	0	0	24.311	0.215	
49	SolarisVille	Drone	Solar Flare	Stable	Thursday	3	66.6029	0	0	33.3014	0.1344	
50	AquaCity	Flying Car	Snowy	Recession	Wednesday	18	68.7717	1	0	82.526	0.3613	
51	Neuroburg	Autonomous Vehicle	Clear	Recession	Saturday	7	106.5692	0	0	96.8811	0.0514	
52	SolarisVille	Autonomous Vehicle	Solar Flare	Stable	Thursday	6	64.5277	0	0	51.6222	0.1454	

Project objectives:

The main objectives of the "Urban Traffic Densities in Cities" dataset are likely to revolve around improving our understanding of urban transportation dynamics and informing decision-making processes aimed at creating more efficient, safe, and sustainable transportation systems in urban areas.

1. **Recognizing Traffic Patterns:** It's possible that the dataset was created to shed light on the density and patterns of traffic in urban regions. This can entail examining changes in traffic density according to the city, the day of the week, the time of day, and other variables.
2. **Management of Traffic:** The dataset was created to aid in the management and optimization of traffic flow in metropolitan settings. Transportation authorities might pinpoint areas of high traffic congestion, adjust the timing of traffic signals, and create plans to reduce traffic during rush hour by evaluating the data.

3. **Infrastructure Planning:** The dataset might be used to help with transportation-related urban planning decisions. The information might be used by planners to identify high-traffic locations and rank the importance of building additional roads, public transportation routes, or other infrastructure changes.
4. **Transportation Policy Development:** The dataset might be used by policymakers to create rules and laws that would enhance urban transportation networks. The data could help with decisions regarding investing in public transportation, increasing the use of alternate ways of flexibility, or implementing congestion pricing schemes.
5. **Safety and Environmental Impact:** The relationship between traffic density, road safety, and the environment might be studied using the dataset. Through an understanding of the relationship between traffic density and air quality, noise pollution, and accident rates, officials may develop interventions focused on increasing road safety and reducing the environmental effect of urban traffic.
6. **Research and Analysis:** The dataset can be used by analysts and researchers to investigate various aspects of urban transportation. This could include creating models to forecast traffic densities in various scenarios, analyzing connections between different variables, and assessing trends over time.

Business problems associated:

The dataset "Urban Traffic Densities in Cities" can provide insights into various business problems associated with urban traffic densities. Some of the key business problems that stakeholders might face include:

Congestion is brought on by high traffic concentrations, which prolongs commuter and transit times. Businesses that depend on on-time deliveries and staff that commute may find this challenging, leading to higher operating expenses and lower productivity. Problems with accessibility reduce foot traffic to restaurants, retail stores, and other facilities in crowded regions, which affects revenue and sales. Businesses that depend on effective transportation networks experience operational disruptions due to demand on transportation infrastructure, and environmental concerns lead to regulatory pressure for sustainable practices. Congested roads raise safety hazards, which might result in collisions and legal consequences. Companies take traffic density into account while choosing locations in order to reduce disturbances and improve accessibility. Data-driven decision-making for strategic planning and operational efficiency optimization is made possible by utilizing urban traffic data.

Data Visualization and Prediction Techniques:

Data Visualization Techniques that will be further used are:

1. **Stacked bar charts**
2. **Bar Charts**
3. **Scatter Plots**

4. **Heatmaps**
5. **Pie charts**

Prediction Techniques that will be used are:

1. **Linear Regression**
2. **Decision Trees**
3. **Gradient-Boost and adaboost (Adaptive Boosting)**

Pre-processing Steps:

Preprocessing steps are vital for ensuring that the "Urban Traffic Densities in Cities" dataset is properly prepared for analysis and modeling. Below are the necessary preprocessing steps:

1. **Handling Missing Data:** Identifying and addressing missing values using techniques like imputation for numerical variables and the most frequent category for categorical ones.
2. **Outlier Detection:** Identifying and managing outliers in numerical variables to prevent them from affecting analysis and model performance.
3. **Feature Engineering:** Creating new features or extracting additional information from existing ones to potentially enhance model performance.
4. **Encoding Categorical Variables:** Converting categorical variables into numerical format using methods like one-hot encoding or label encoding.
5. **Feature Scaling:** Scaling numerical features to ensure that features with different scales do not dominate the model.
6. **Data Transformation:** Transforming variables, if necessary, to meet the assumptions of the chosen modeling technique.
7. **Handling Imbalanced Data:** Addressing class imbalance in the target variable, if present, using techniques such as oversampling or undersampling.
8. **Dataset Splitting:** Dividing the dataset into training and testing sets to evaluate model performance.
9. **Normalization:** Normalizing the data, if required by the algorithm being used.
10. **Dimensionality Reduction:** Reducing the dataset's dimensionality using techniques like Principal Component Analysis (PCA) to improve computational efficiency and reduce overfitting.

By following these preprocessing steps, our dataset is ready for analysis and modeling, leading to more accurate and reliable insights into urban traffic densities in cities.

Dataset Summary:

The dataset provides detailed information about traffic conditions in a futuristic urban environment, comprising over 10 lakhs records. Each record represents a snapshot of various factors influencing traffic conditions in six fictional cities.

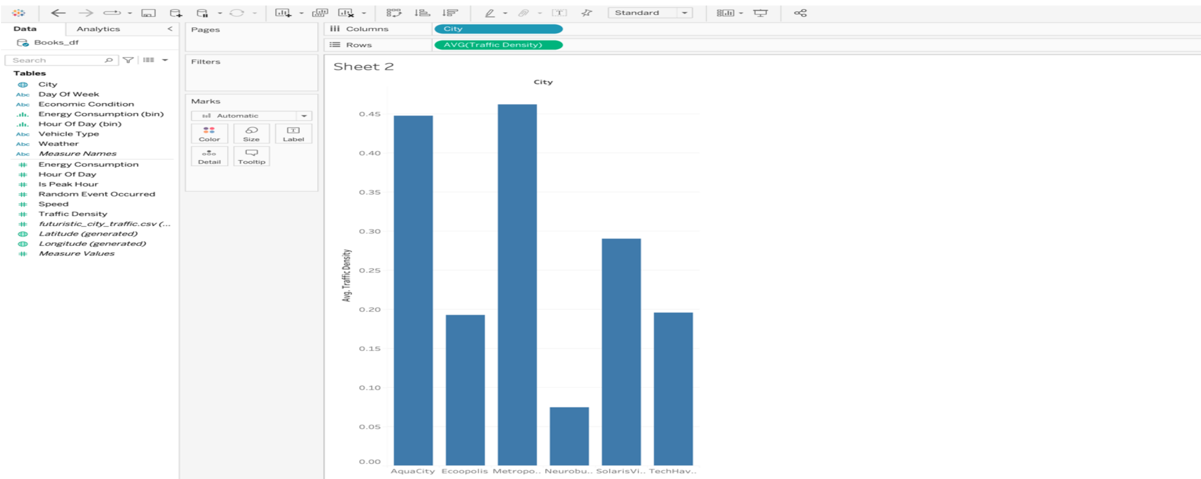
Features include:

- City: Name of the city (e.g., MetropolisX, SolarisVille).
- Vehicle Type: Type of vehicle (e.g., Car, Flying Car).
- Weather Conditions: Current weather (e.g., Clear, Rainy).
- Economic Conditions: Economic state of the city (e.g., Booming, Recession).
- Day of Week: Day of the week.
- Hour of Day: Hour of the day when the data was recorded.
- Speed: Recorded vehicle speed.

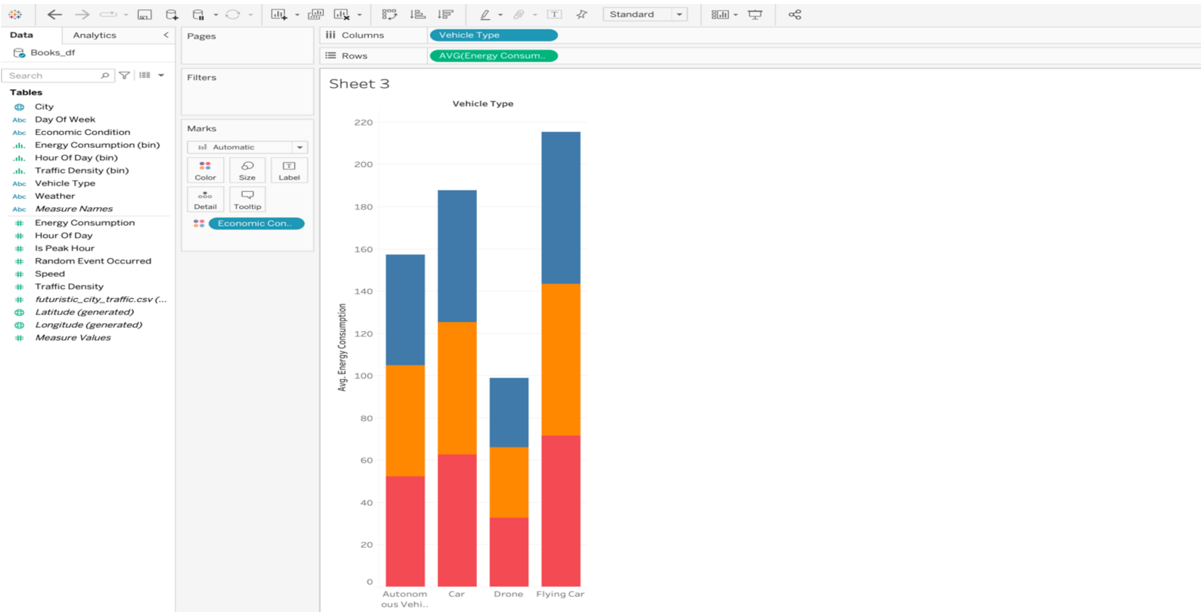
- Energy Consumption: Estimated energy consumption based on vehicle type and speed.
- Is Peak Hour: Indicator if the record was during peak traffic hours.
- Random Event Occurred: Indicator if a random event (e.g., accidents, road closures) occurred.
- Traffic Density: Density of traffic at the time of recording.

The dataset aims to provide insights into traffic patterns, dynamics, and factors affecting traffic densities in urban areas. It is comprehensive and diverse, covering various aspects of traffic conditions and urban environments. The dataset is suitable for exploratory data analysis, modeling, and developing strategies for traffic management, urban planning, and transportation policies.

Predictions:



This bar chart shows the relationship between city and average traffic density.



This stacked bar chart shows the relationship between vehicle type and average energy consumption.

Conclusion for the Proposal:

In summary, the dataset on "Urban Traffic Densities in Cities" offers a valuable resource for comprehending and analyzing traffic situations in advanced urban settings. With a vast collection of over 10 lakhs entries covering diverse factors impacting traffic density across six fictional cities, the dataset provides rich insights into the intricacies of urban traffic.

By exploring variables such as city names, vehicle types, weather conditions, economic states, days of the week, time of day, vehicle speeds, energy consumption, peak traffic hours, and random incidents, stakeholders can gain a thorough understanding of traffic patterns, trends, and obstacles.

Given its extensive scope and depth, the dataset is apt for conducting exploratory analyses, building models, and crafting strategies to tackle traffic management, urban infrastructure planning, environmental sustainability, and economic efficiency in urban locales.

To sum up, the dataset on "Urban Traffic Densities in Cities" holds considerable promise for informing data-driven decision-making and fostering innovation in urban transportation systems, leading towards safer, more effective, and sustainable urban environments in the future.