Data Collection and Preprocessing Phase

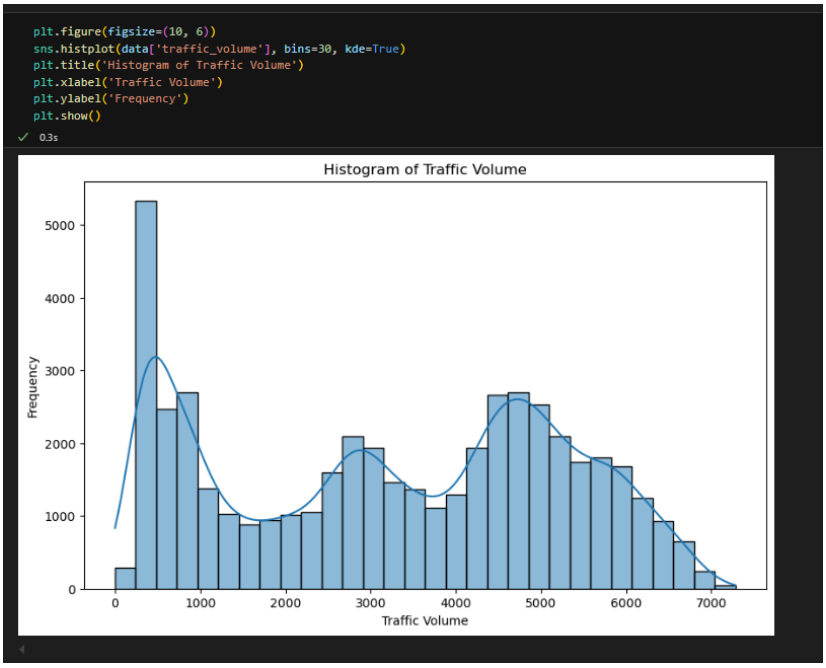
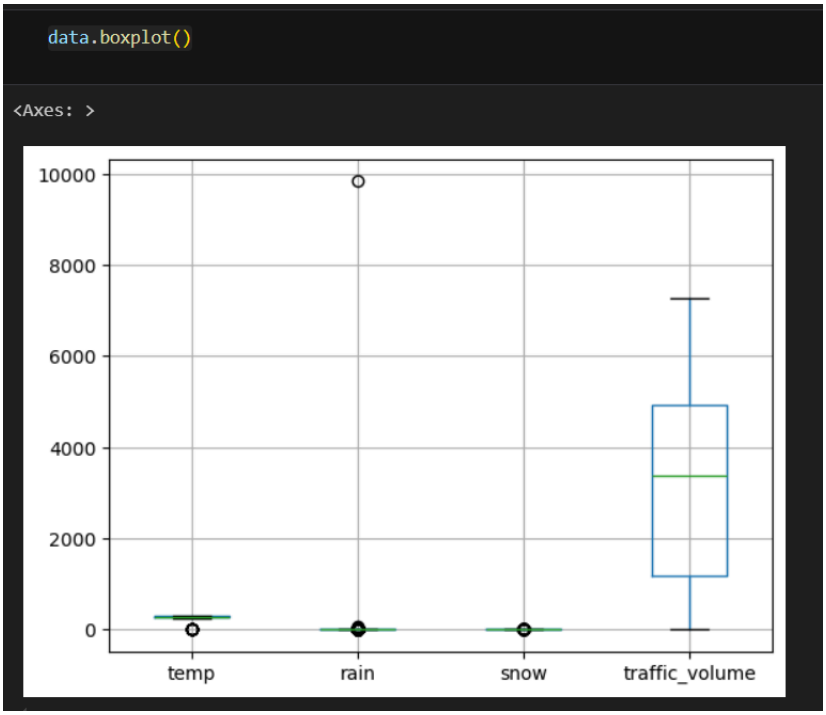
Date	12th October 2024
Team ID	LTVIP2024TMID24968
Project Title	TrafficTelligence Advanced Traffic Volume Estimation with Machine Learning
Maximum Marks	6 Marks

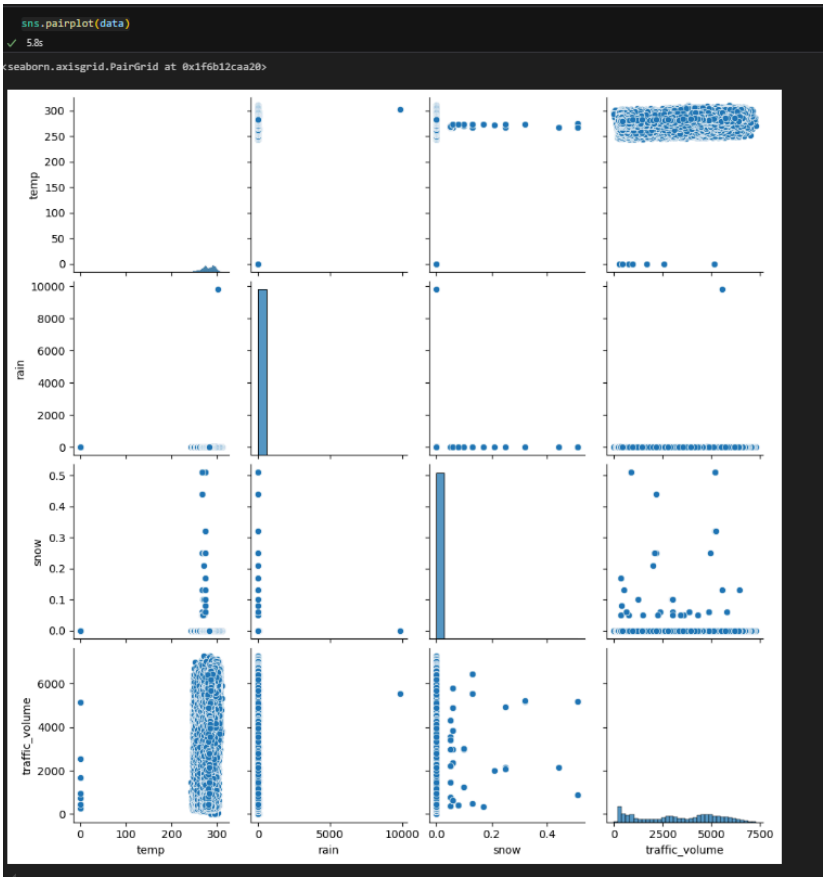
Data Exploration and Preprocessing Template

Dataset variables will undergo statistical analysis to uncover patterns and identify outliers, utilizing Python for preprocessing tasks such as normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring high-quality data for subsequent analysis and modeling, ultimately establishing a solid foundation for insights and accurate predictions.

Section	Description
Data Overview	`data.describe()` [8]

	temp	rain	snow	traffic_volume
count	48151.000000	48202.000000	48192.000000	48204.000000
mean	281.205351	0.334278	0.000222	3259.818355
std	13.343675	44.790062	0.008169	1986.860670
min	0.000000	0.000000	0.000000	0.000000
25%	272.160000	0.000000	0.000000	1193.000000
50%	282.460000	0.000000	0.000000	3380.000000
75%	291.810000	0.000000	0.000000	4933.000000
max	310.070000	9831.300000	0.510000	7280.000000

Univariate Analysis	
Bivariate Analysis	
Multivariate Analysis	Pairplot


sns.pairplot(data)
5.8s
<seaborn.axisgrid.PairGrid at 0x1f6b12caa20>
temp
rain
snow
traffic_volume

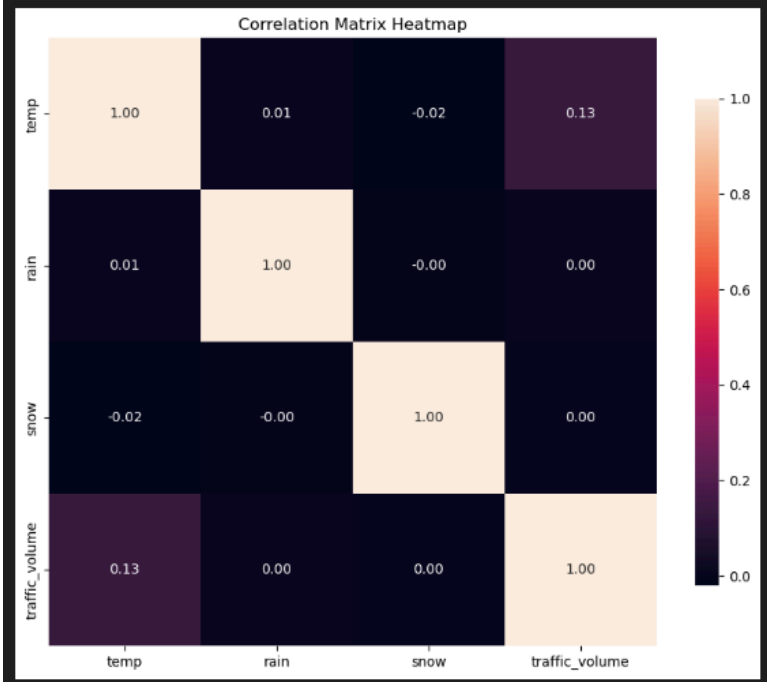
Heat Map

<table>
<tr><td></td><td>

```
numeric_df = data.select_dtypes(include=[np.number])
corr = numeric_df.corr()
corr
```

	temp	rain	snow	traffic_volume
temp	1.000000	0.009069	-0.019760	0.130051
rain	0.009069	1.000000	-0.000090	0.004714
snow	-0.019760	-0.000090	1.000000	0.000733
traffic_volume	0.130051	0.004714	0.000733	1.000000

```
plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True, fmt=".2f", square=True, cbar_kws={"shrink": .8})
plt.title('Correlation Matrix Heatmap')
plt.show()
```



</td></tr>
<tr><td>Outliers and Anomalies</td><td>There are no outliers or anomalies in the dataset.</td></tr>
<tr><td colspan="2">Data Preprocessing Code Screenshots</td></tr>
</table>

Loading Data

```
data = pd.read_csv('traffic volume.csv')
```
✓ 0.0s

```
data
```
✓ 0.0s

	holiday	temp	rain	snow	weather	date	Time	traffic_volume
0	0	288.28	0.0	0.0	1.0	02-10-2012	09:00:00	5545
1	0	289.36	0.0	0.0	1.0	02-10-2012	10:00:00	4516
2	0	289.58	0.0	0.0	1.0	02-10-2012	11:00:00	4767
3	0	290.13	0.0	0.0	1.0	02-10-2012	12:00:00	5026
4	0	291.14	0.0	0.0	1.0	02-10-2012	13:00:00	4918
...	...	...	...	...	...	...	...	...
48199	0	283.45	0.0	0.0	1.0	30-09-2018	19:00:00	3543
48200	0	282.76	0.0	0.0	1.0	30-09-2018	20:00:00	2781
48201	0	282.73	0.0	0.0	Thunderstorm	30-09-2018	21:00:00	2159
48202	0	282.09	0.0	0.0	1.0	30-09-2018	22:00:00	1450
48203	0	282.12	0.0	0.0	1.0	30-09-2018	23:00:00	954

48204 rows × 8 columns

Handling Missing Data

```
data['temp'] = data['temp'].fillna(data['temp'].median())
data['rain'] = data['rain'].fillna(data['rain'].median())
data['snow'] = data['snow'].fillna(data['snow'].median())
```
✓ 0.0s

```
print(data['weather'].value_counts())
```
✓ 0.0s

```
weather
Clouds          15144
Clear           13383
Mist             5942
Rain             5665
Snow             2875
Drizzle          1818
Haze             1359
Thunderstorm     1033
Fog               912
Smoke              20
Squall              4
Name: count, dtype: int64
```

```
data['weather'].fillna('Clouds',inplace=True)
```
✓ 0.0s

```
C:\Users\saisa\AppData\Local\Temp\ipykernel_15312\1731991201.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  data['weather'].fillna('Clouds',inplace=True)
```

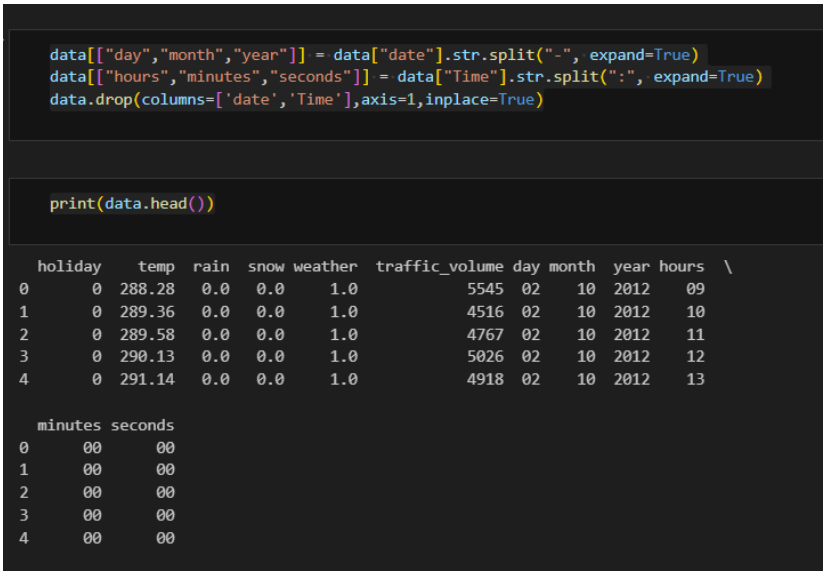
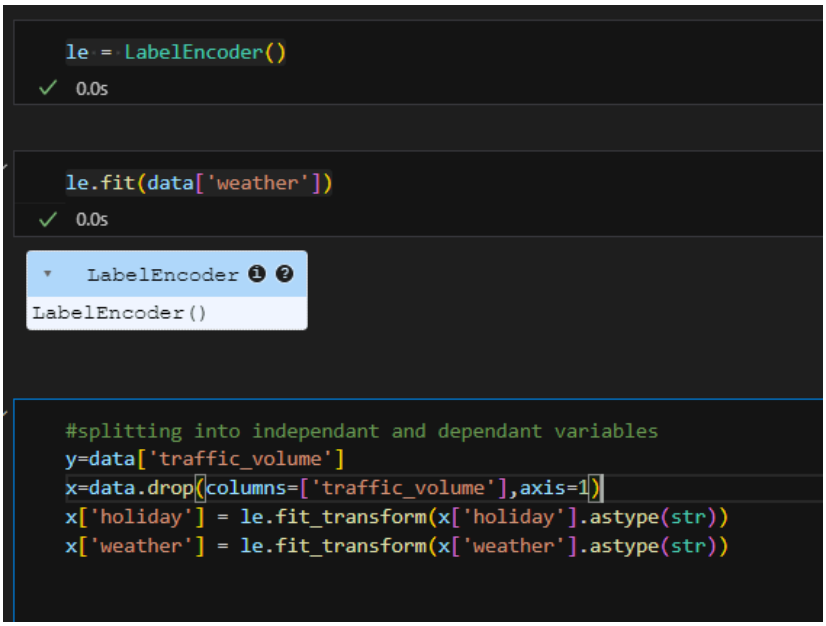
```
# prompt: print NaN in holiday column to 0

data['holiday'].fillna(0, inplace=True)
```
✓ 0.0s

```
C:\Users\saisa\AppData\Local\Temp\ipykernel_15312\821488972.py:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  data['holiday'].fillna(0, inplace=True)
```

Data Transformation	
Feature Engineering	
Save Processed Data