Traffic Intelligence: Advanced Traffic Volume Estimation with Machine Learning

MENTOR*: *SRI L.LAKSHMI NARAYANA

TEAM ID*: *LTVIP2025TMID40409

TEAM MEMBER:

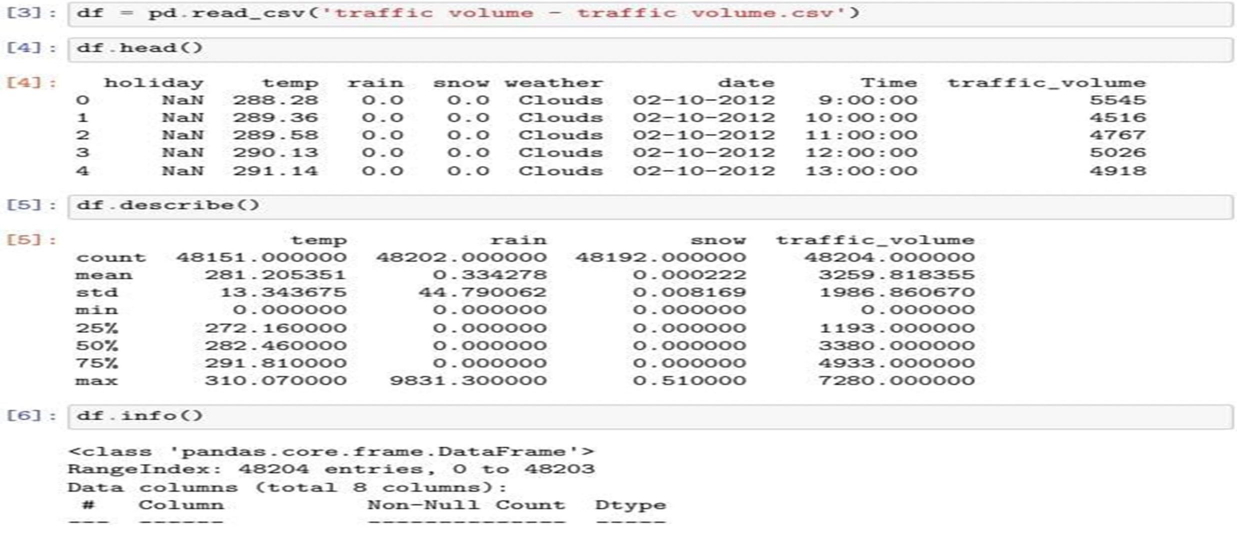
1) **MARIPI SAI**

MAIL ID: saimaripi3@gmail.com

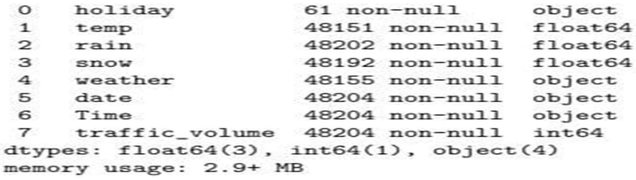
MY TASKS:

***TSK-338836*:**

Analyse the data



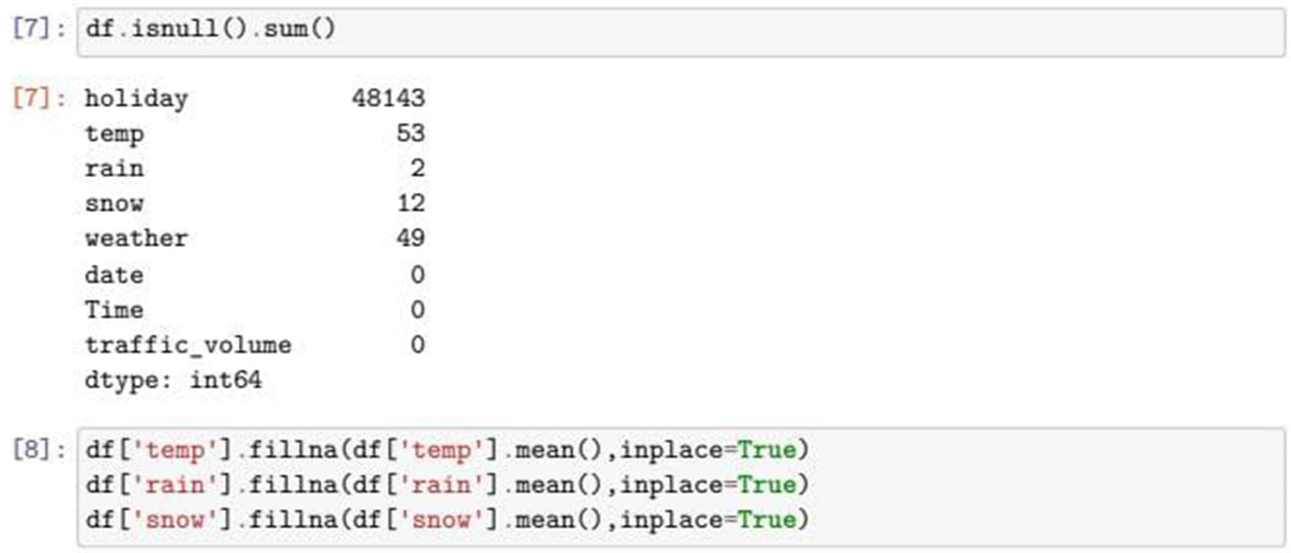
```
[3]: df = pd.read_csv('traffic volume - traffic volume.csv')

[4]: df.head()

[4]:   holiday    temp  rain  snow weather        date      Time  traffic_volume
     0     NaN  288.28   0.0   0.0  Clouds  02-10-2012   9:00:00            5545
     1     NaN  289.36   0.0   0.0  Clouds  02-10-2012  10:00:00            4516
     2     NaN  289.58   0.0   0.0  Clouds  02-10-2012  11:00:00            4767
     3     NaN  290.13   0.0   0.0  Clouds  02-10-2012  12:00:00            5026
     4     NaN  291.14   0.0   0.0  Clouds  02-10-2012  13:00:00            4918

[5]: df.describe()

[5]:              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

[6]: df.info()

    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 48204 entries, 0 to 48203
    Data columns (total 8 columns):
     #   Column          Non-Null Count  Dtype
    ---  ------          --------------  -----
```



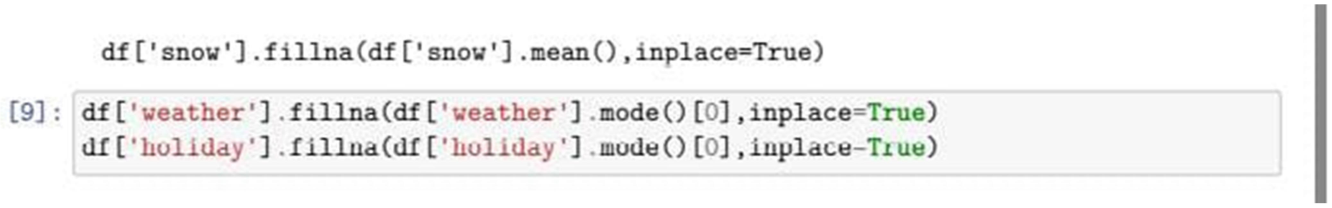
```
     0   holiday         61 non-null     object
     1   temp            48151 non-null  float64
     2   rain            48202 non-null  float64
     3   snow            48192 non-null  float64
     4   weather         48155 non-null  object
     5   date            48204 non-null  object
     6   Time            48204 non-null  object
     7   traffic_volume  48204 non-null  int64
    dtypes: float64(3), int64(1), object(4)
    memory usage: 2.9+ MB
```

***TSK-338837*:**

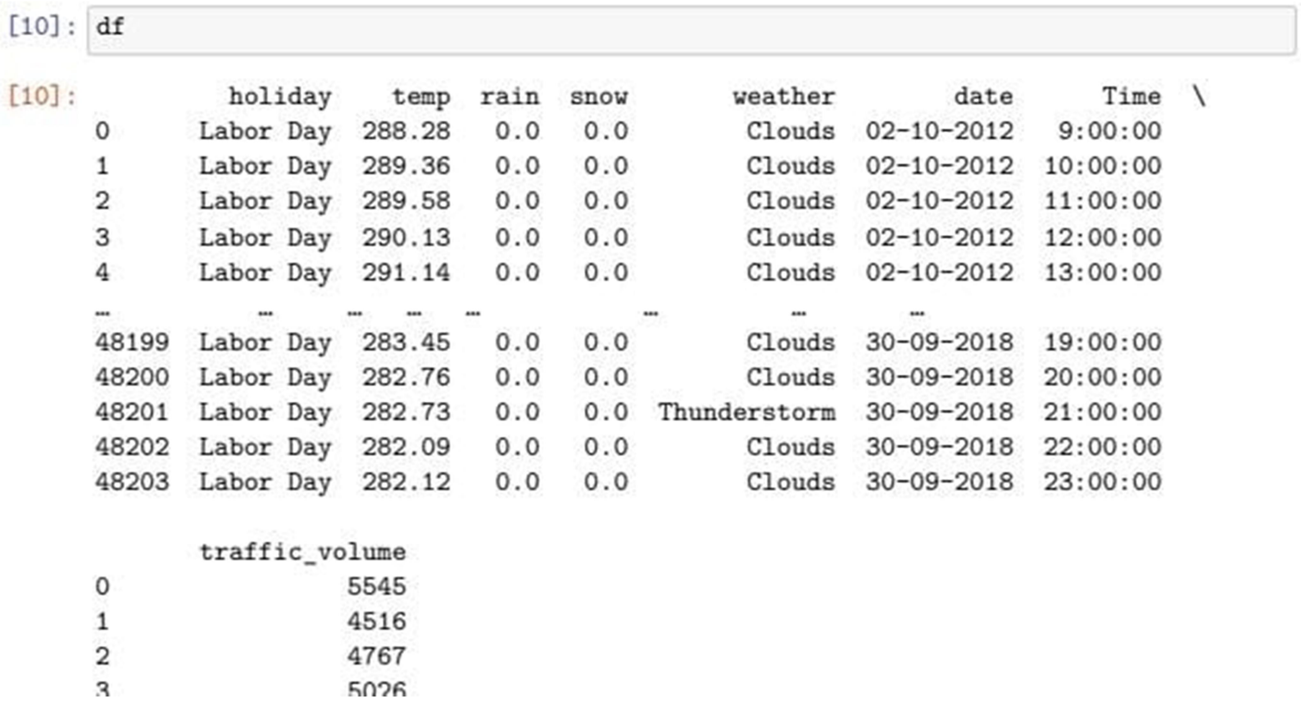
Handling Missing Values



```
[7]: df.isnull().sum()

[7]: holiday           48143
     temp                 53
     rain                  2
     snow                 12
     weather              49
     date                  0
     Time                  0
     traffic_volume        0
     dtype: int64

[8]: df['temp'].fillna(df['temp'].mean(),inplace=True)
     df['rain'].fillna(df['rain'].mean(),inplace=True)
     df['snow'].fillna(df['snow'].mean(),inplace=True)
```



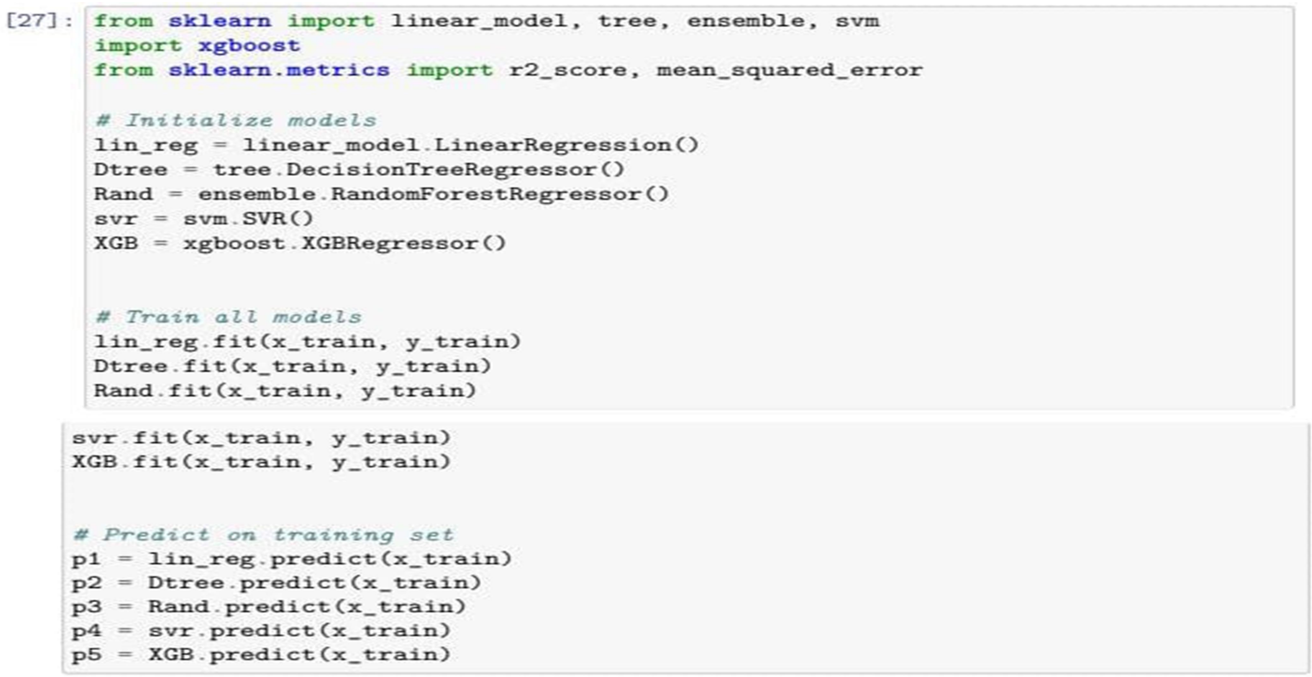
```
      df['snow'].fillna(df['snow'].mean(),inplace=True)

[9]: df['weather'].fillna(df['weather'].mode()[0],inplace=True)
     df['holiday'].fillna(df['holiday'].mode()[0],inplace=True)
```



```
[10]: df

[10]:        holiday    temp  rain  snow       weather        date      Time  \
      0      Labor Day  288.28   0.0   0.0        Clouds  02-10-2012   9:00:00
      1      Labor Day  289.36   0.0   0.0        Clouds  02-10-2012  10:00:00
      2      Labor Day  289.58   0.0   0.0        Clouds  02-10-2012  11:00:00
      3      Labor Day  290.13   0.0   0.0        Clouds  02-10-2012  12:00:00
      4      Labor Day  291.14   0.0   0.0        Clouds  02-10-2012  13:00:00
      ...          ...     ...   ...   ...           ...         ...       ...
      48199  Labor Day  283.45   0.0   0.0        Clouds  30-09-2018  19:00:00
      48200  Labor Day  282.76   0.0   0.0        Clouds  30-09-2018  20:00:00
      48201  Labor Day  282.73   0.0   0.0  Thunderstorm  30-09-2018  21:00:00
      48202  Labor Day  282.09   0.0   0.0        Clouds  30-09-2018  22:00:00
      48203  Labor Day  282.12   0.0   0.0        Clouds  30-09-2018  23:00:00

             traffic_volume
      0                5545
      1                4516
      2                4767
      3                5026
```

***TSK-338842*:**

Training and Testing the Model



[27]:

```
from sklearn import linear_model, tree, ensemble, svm
import xgboost
from sklearn.metrics import r2_score, mean_squared_error

# Initialize models
lin_reg = linear_model.LinearRegression()
Dtree = tree.DecisionTreeRegressor()
Rand = ensemble.RandomForestRegressor()
svr = svm.SVR()
XGB = xgboost.XGBRegressor()


# Train all models
lin_reg.fit(x_train, y_train)
Dtree.fit(x_train, y_train)
Rand.fit(x_train, y_train)
svr.fit(x_train, y_train)
XGB.fit(x_train, y_train)


# Predict on training set
p1 = lin_reg.predict(x_train)
p2 = Dtree.predict(x_train)
p3 = Rand.predict(x_train)
p4 = svr.predict(x_train)
p5 = XGB.predict(x_train)
```