Traffic Prediction using Machine Learning

1. What kind of preprocessing was done to the dataset before training the model?

- **Data Encoding**: Categorical variables such as `Day` and `Zone` were encoded using numerical codes. For instance, `CodedDay` was used to represent days in a numerical format.
- **Data Cleaning**: The dataset was cleaned to remove any inconsistencies or missing data.
- **Feature Selection**: Features like `Weather`, `Temperature`, `Day`, and `Zone` were used as input variables, and the target variable was `Traffic`.

2. Which machine learning algorithm was used to predict traffic, and why was it chosen?

1. The notebook doesn't explicitly mention the algorithm used, but based on the steps and the type of output (continuous values for traffic), it is likely a **regression algorithm** (possibly **Linear Regression** or **Random Forest Regression**).

2. Why Chosen: Regression models are well-suited for predicting continuous values, such as traffic levels, based on numerical and categorical inputs.

3. How were categorical variables (like Day and Zone) encoded for model input?

1. Numerical Encoding: Categorical variables such as `Day` and `Zone` were encoded into numerical representations. For example, `CodedDay` was used instead of the actual day names, and `Zone` numbers were used directly.

2. This encoding converts categorical data into a format suitable for machine learning models.

4. What metric was used to evaluate the model performance, and how is it calculated?

1. The evaluation metric used is the **error rate**. It is calculated as:

2. Error = (ypred−ytest) / ytest×100

3. where `y_pred` is the predicted traffic value, and `y_test` is the actual traffic value.

5. What does the calculated error rate of 12.16% indicate about the model's performance?

1. The error rate of **12.16%** indicates that, on average, the predicted traffic differs from the actual traffic by about 12.16%. This error rate shows that the model is reasonably accurate, but there is still room for improvement.

2. A lower error rate would indicate a more precise model.

6. How was the dataset split between training and testing? Was cross-validation used?

1. The dataset was likely split into training and testing sets using a method like `train_test_split`. However, cross-validation is not explicitly mentioned in the notebook.

2. Cross-validation, if used, would provide a more robust evaluation of the model by splitting the data into multiple training and test sets.

7. What are the potential limitations or improvements that can be made to increase model accuracy?

- **Limitations**:

1. The model might not capture complex patterns in the data if only a basic algorithm like linear regression was used.

2. The dataset may not account for all relevant features (e.g., real-time traffic events, road conditions).

- **Improvements**:

  1. Using a more complex model like **Random Forest** or **Gradient Boosting** could improve accuracy.

  2.Applying **hyperparameter tuning** and **feature engineering** (e.g., creating lag variables for time-series data) can enhance performance.

  3.Implementing **cross-validation** would provide a more reliable evaluation of the model.