**Ideation Phase**

**Brainstorm & Idea Prioritization Template**

|  |  |
| --- | --- |
| Date | 16 July 2025 |
| Team ID | LTVIP2025TMID44001 |
| Project Name | TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning |
| Maximum Marks | 4 Marks |

**Brainstorm & Idea Prioritization Template:**

Traffic congestion in urban cities leads to delays, pollution, and safety issues. This project aims to **predict traffic volume using weather and temporal data** with machine learning models, enabling smart traffic planning and forecasting.

**Step-1: Team Gathering, Collaboration and Select the Problem Statement**

In the initial phase of the project, we formed a collaborative and motivated team with shared interests in data science, machine learning, and smart city applications. Each team member brought unique skills to the table, including:

* Data preprocessing and analysis
* Machine learning model development
* Visualization and interpretation
* Web development and deployment

Team collaboration was facilitated through online platforms such as Google Meet, WhatsApp, and GitHub to ensure regular updates, knowledge sharing, and task distribution.

**Selecting the Problem Statement**

After a brainstorming session and reviewing the problem statements provided by SmartInternz, we selected the following:

**Problem Statement:**  
“Build a Machine Learning model to estimate traffic volume using weather and temporal features such as temperature, rain, snow, holidays, and date-based information (day, month, year).”

This problem aligns perfectly with real-world smart city challenges and offers a practical application of machine learning in optimizing urban mobility and planning.

**Step-2: Brainstorm, Idea Listing and Grouping**

Here’s a list of ideas generated during the brainstorming phase:

1. Use regression models (Linear, Decision Tree, Random Forest, SVR, XGBoost)
2. Perform data cleaning and handle missing values
3. Use LabelEncoder for categorical data (holiday, weather)
4. Normalize data using StandardScaler
5. Extract day, month, and year from the date
6. Compare model performance using R², MAE, RMSE
7. Save trained model and encoders using pickle
8. Build a user interface using Flask for prediction
9. Visualize feature importance to improve interpretability
10. Add threshold-based alerts for extreme traffic volume *(optional)*
11. We grouped similar ideas into functional categories:

| **Category** | **Ideas Grouped** |
| --- | --- |
| **Data Preprocessing** | Handle null values, label encoding, scaling, date feature extraction |
| **Model Development** | Train and compare multiple ML regression models |
| **Evaluation & Validation** | Use R², RMSE, MAE for assessing accuracy |
| **Deployment** | Save model and scalers using pickle for web app integration |
| **User Interaction (UI)** | Flask app for inputs and results |
| **Visualization & Insights** | Feature importance plot, traffic volume patterns |
| **Advanced Features (Optional)** | Smart alerts based on prediction thresholds |

**Step-3: Idea Prioritization**

| S.No | Idea | Feasibility (1–5) | Impact (1–5) | Priority Score | Priority Level |
| --- | --- | --- | --- | --- | --- |
| 1 | Apply ML models (LR, RF, XGB, etc.) | 5 | 5 | 25 | High |
| 2 | Handle missing values, encode & scale features | 5 | 4 | 20 | High |
| 3 | Extract day, month, year from date | 5 | 3 | 15 | Medium |
| 4 | Compare model performance using R², RMSE | 4 | 5 | 20 | High |
| 5 | Save model & encoders using pickle | 5 | 4 | 20 | High |
| 6 | Flask web app for prediction UI | 3 | 5 | 15 | Medium |
| 7 | Feature importance visualization | 3 | 3 | 9 | Low |
| 8 | Smart alerts based on threshold (Optional) | 2 | 3 | 6 | Low |