SPRINT 2 DOCUMENT

SPACE TRAFFIC DENSITY PREDICTION

Team Members:
Anushka Dutta
Himaja S
Karanbir Singh
Shravani R S
Tarunaa A C

1. Problem Statement

As space becomes increasingly crowded with satellites, space stations, and debris, managing traffic has become a significant challenge. Predicting space traffic density is essential for ensuring safe and efficient operations in space. This project aims to address the following key challenges:

- Orbital Congestion: Low Earth Orbit (LEO), Medium Earth Orbit (MEO), and Geostationary Earth Orbit (GEO) have varying levels of traffic density.
- Object Diversity: Different objects like satellites, space stations, and debris contribute differently to space traffic density.
- Dynamic Patterns: Traffic density changes with time, influenced by dynamic orbital behaviors and operational schedules.

The project's objective is to create a predictive model that estimates space traffic density in real-time, aiding decision-making in satellite operations, launch planning, and collision avoidance.

2. Scope

The second sprint focused on the following:

1. Feature Engineering: Implementation of preprocessing techniques for data readiness.
2. Model Development and Evaluation: Developed and evaluated multiple machine learning models, including linear regression, regularization models, and ensemble-based techniques. Explored time-series models (ARIMA, SARIMAX) and deep learning architectures (RNN, GRU, LSTM).
3. Performance Assessment: Measuring model effectiveness using MSE and R^2 metrics.
4. Challenges Identification: Highlighting potential data and modeling gaps.

3. Deliverables and Outcomes

3.1 Feature Engineering

Enhancing model performance by transforming raw data into a suitable format for machine learning.

3.1.1 Temporal Features:

- Hour of the Day: Extracted from the Timestamp column to capture daily traffic patterns such as peak and off-peak hours.
- Day of the Week: Derived from the Timestamp to identify weekday versus weekend traffic variations.
- Day of the Month: Extracted to detect monthly trends and periodic traffic behavior.

3.1.2 Peak Time Transformation:

The Peak_Time column was converted into an integer representing the hour, ensuring it is treated as a numerical feature for model compatibility.

3.1.3 Categorical Features:

Location and Object_Type: Categorical variables were encoded using OneHotEncoder to avoid ordinal misinterpretation while capturing distinct categories.

3.1.4 Target Variable Scaling:

The Traffic_Density target variable was scaled using StandardScaler to standardize its range, improving the model's performance.

3.1.5 Combined Transformation:

A ColumnTransformer was used to integrate categorical and numerical transformations into a single pipeline, ensuring consistency and efficiency in preprocessing.

3.2 Model Performance

The table below summarizes the key results from the models trained and evaluated during Sprint 2:

Model	Mean Squared Error (MSE)	R^2 Score	Remarks
Linear Regression	1.0513	-0.03	Underperforming; R^2 indicates worse performance than predicting the mean value.
Cross-Validation (Linear)	-1.0111 (CV MSE)	N/A	Negative CV MSE suggests model/data incompatibility. Test size 0.3 had the lowest MSE (0.9811).
Ridge Regression	1.0510	-0.03	Marginal improvement; fails to capture underlying data patterns.
Lasso Regression	0.9664	-0.003	Slight MSE improvement; still underperforms with minimal variance capture.
KNN Regressor	1.2181	-0.26	High MSE and negative R^2 indicate poor fit and failure to capture meaningful patterns.
Random Forest	1.2211	-0.02	Overfitting observed; performs poorly on unseen data. Cross-validated R^2: -0.098.
Gradient Boosting	1.1933	-0.17	Better than Random Forest but still suboptimal. Cross-validated R^2: -0.081.
ARIMA	791.44	-0.006	Minimal variance capture; high error suggests insufficient temporal pattern recognition.

SARIMAX	803.83	-0.022	Similar to ARIMA; struggles to align predictions with actual values.
RNN	1398.91	-0.78	Severe underfitting; ineffective pattern capture.
GRU	1120.66	-0.42	Slight improvement over RNN; still suboptimal.
LSTM	1832.86	-1.33	Significant underfitting; performs worse than a mean predictor.

4. Key Observations

4.1 Model Performance

- Consistent Underperformance:
  Most models showed negative R^2 scores and high Mean Squared Errors (MSE), indicating they are not effectively capturing the patterns in the data.

4.2 Underfitting

- Models, including linear regression, ensemble models, and neural networks, consistently underperformed, suggesting that they are not complex enough or not well-suited to the problem at hand.

4.3 Data Challenges

- Feature Relevance: The current feature set may lack complexity or relevance to effectively model traffic density.
- Quality Concerns: The poor results across various algorithms suggest the dataset may require better preprocessing, feature selection, or more relevant data to improve model predictions.

4.4 Time-Series and Neural Networks

- ARIMA and SARIMAX: Poor results suggest insufficient temporal pattern recognition or inappropriate model parameters.

- RNN, GRU, and LSTM: Underfitting was observed, likely due to inadequate preprocessing or suboptimal hyperparameters.

4.5 Ensemble Models

- Random Forest and Gradient Boosting: These models showed marginal improvement but still failed to generalize effectively.
- High MSE values suggest that better feature engineering and tuning are required.

PseudoCode:

1. Feature Engineering:
    - Step 1: Convert Timestamp to Datetime Format
    - Step 2: Extract Hour from Peak_Time
    - Step 3: Extract Features from Timestamp
    - Step 4: Define Features (X) and Target (y)
    - Step 5: Preprocess Features
    - Step 6: Fit and Transform Features
    - Step 7: Scale the Target Variable
2. Linear Regression:
    - Step 1: Split the Data
    - Step 2: Initialize and Train the Model
    - Step 3: Make Predictions
    - Step 4: Evaluate the Model
    - Step 5: Output Results
3. Ridge Regression:
    - Step 1: Initialize the Ridge Regression Model
    - Step 2: Split Data into Training and Testing Sets
    - Step 3: Train the Ridge Regression Model
    - Step 4: Predict on Test Data
    - Step 5: Evaluate the Ridge Regression Model
    - Step 6: Perform Cross-Validation
    - Step 7: Output Results
4. Lasso Regression:
    - Step 1: Initialize the Lasso Regression Model
    - Step 2: Split Data into Training and Testing Sets

- Step 3: Train the Lasso Regression Model
- Step 4: Predict on Test Data
- Step 5: Evaluate the Lasso Regression Model
- Step 6: Perform Cross-Validation
- Step 7: Output Results

5. KNN Regressor:
    - Step 1: Initialize the KNN Regressor Model
    - Step 2: Split Data into Training and Testing Sets
    - Step 3: Train the KNN Regressor
    - Step 4: Predict on Test Data
    - Step 5: Evaluate the KNN Regressor
    - Step 6: Perform Cross-Validation
    - Step 7: Output Results
6. Random Forest Regressor:
    - Step 1: Import and Initialize the Random Forest Regressor
    - Step 2: Fit the Model on Training Data
    - Step 3: Predict on Test Data
    - Step 4: Evaluate the Model
    - Step 5: Cross-Validation
    - Step 6: Output Results
7. Gradient Boosting Regressor:
    - Step 1: Import and Initialize the Model
    - Step 2: Fit the Model
    - Step 3: Predict on Test Data
    - Step 4: Evaluate the Model
    - Step 5: Perform Cross-Validation
    - Step 6: Output Results
8. ARIMA Model:
    - Step 1: Fit ARIMA Model
    - Step 2: Forecast
    - Step 3: Evaluate Model
    - Step 4: Visualization
9. SARIMA Model:
    - Univariate Data Selection
    - Stationarity Check
    - Train-Test Split
    - SARIMA Model Configuration

- Fit the SARIMA Model
- Forecasting
- Model Evaluation
- Visualization

10. Recurrent Neural Network, Gated Neural Network and Long Short Term Memory:
    - Model Setup
    - Model Compilation
    - Model Training
    - Predictions
    - Evaluation

Next Steps:

- Model Improvement
- Model Deployment

Conclusion:

Despite challenges with model performance in Sprint 2, significant groundwork has been laid for feature engineering and model exploration. The results underscore the importance of refining data and optimizing models to align with project objectives. Sprint 3 will focus on addressing the identified issues to achieve better predictive accuracy and actionable insights.