TrafficTelligence-Advanced-Traffic-Volume-Estimation-with-Machine-Learning

The Traffic Intelligence project aims to revolutionize traffic management through the implementation of advanced machine learning techniques for accurate and real-time traffic volume estimation.

Project Overview

This project predicts traffic volume using historical data and machine learning techniques. The dataset includes weather conditions, temporal data, and other features that influence traffic. By preprocessing the data and training multiple machine learning models, the project identifies the most effective model for accurate predictions.

Table of Contents

Features

The dataset includes the following features:

- **Temporal Data**: Date and Time split into day, month, year, hours, minutes, and seconds.
- **Weather Conditions**: Temperature, rainfall, snowfall, and general weather categories.
- **Traffic Volume**: The target variable indicating the number of vehicles.
- **Holidays**: Categorical feature indicating whether the date is a holiday.

Technologies Used

- **Python Libraries**:
    - `pandas`, `numpy`: Data manipulation and numerical computations.
    - `seaborn`, `matplotlib`: Data visualization.
    - `scikit-learn`: Machine learning models and utilities.
    - `xgboost`: Gradient boosting model.
    - `pickle`: Model serialization.

Setup

1. Clone the repository.
2. Install the required Python libraries:

    ```
    pip install pandas numpy seaborn matplotlib scikit-learn xgboost
    ```
3. Place the dataset (`traffic volume.csv`) in the project directory.
4. Run the main script to preprocess the data, train models, and evaluate results.

Steps in the Pipeline

1. Data Preprocessing

- Load the dataset.
- Handle missing values:
    - Replace numeric nulls with mean values.
    - Replace categorical nulls with the most frequent value (`'Clouds'`).
- Split `date` and `Time` columns into individual components.
- Encode categorical columns (`weather` and `holiday`) using `LabelEncoder`.
- Standardize numerical features.

2. Exploratory Data Analysis

- Analyze feature correlations using a heatmap.
- Visualize feature distributions and relationships using:
    - Count plots.
    - Pair plots.
    - Box plots.

3. Model Training

Train the following models:

1. Linear Regression
2. Decision Tree Regressor

3. Random Forest Regressor
4. Support Vector Regressor (SVR)
5. XGBoost Regressor

4. Model Evaluation

Evaluate models using:

- **R-squared Score**: Measures prediction accuracy.
- **Root Mean Squared Error (RMSE)**: Measures average error in predictions.

5. Deployment

- Save the best-performing model (`Random Forest Regressor`) using `pickle` for future predictions.
- Save the `LabelEncoder` for encoding new categorical data.

Model Evaluation

The models were compared based on R-squared scores and RMSE. The `Random Forest Regressor` outperformed the others with the lowest RMSE, making it the chosen model for deployment.

Deployment

The best-performing model and label encoder are saved as:

- `model.pkl`: Serialized model file.
- `encoder.pkl`: Serialized encoder file.

These files can be loaded for predictions using unseen data.

Visualizations

1. Correlation Heatmap: Visualizes relationships between features.
2. Pair Plots: Shows pairwise relationships for selected features.
3. Count Plots: Displays distributions of categorical variables.
4. Box Plots: Highlights potential outliers in numeric data.

Future Enhancements

- Incorporate additional features like road conditions or real-time traffic updates.
- Experiment with hyperparameter tuning to improve model performance.
- Deploy the model as a web application or API for real-time traffic volume predictions.

Authors

- **Pamuru Bhagyasree** : Developer and Data Scientist.