Model Building Documentation

Step 1: Data Integration & Identify the Target Column

- Objective: Combine relevant features from multiple datasets into a single feature matrix (X) and define the target variable (y).
- Process:

  - Datasets Loaded:

    - 'processed_logistics_data.csv'

    - 'pca_results (3).csv'

    - 'cleaned_and_engineered_logistics_data.csv'

  - Data Inspection:

    - Displayed summaries of the datasets to understand their structure and content.

  - Feature Matrix (X) and Target Variable (y):

    - Combined features from the datasets, ensuring the target column 'actual_time' is excluded from the feature matrix.

    - Defined 'actual_time' as the target variable.

  - Validation:

    - Verified that all datasets have the same number of rows.

    - Checked for missing values and duplicate rows in the combined dataset.

- Output:

    - Combined Feature Matrix Shape: (227, 135)

    - Target Variable Shape: (227,)

Step 2: Train-Test Splitting

- Objective: Split the dataset into training and testing sets.
- Process:

  - Used an 80-20 split for training and testing.

  - Saved the train-test splits as CSV files for future reference.

- Output:

  - X_train shape: (181, 135), y_train shape: (181,)

  - X_test shape: (46, 135), y_test shape: (46,)

Step 3: Baseline Modeling

- Objective: Establish a baseline model for comparison.
- Process:

  - Model Used: Linear Regression

  - Evaluation Metrics:

  - Mean Absolute Error (MAE)

  - Root Mean Squared Error (RMSE)

  - R^2 Score

- Output:

  - MAE: 0.0019

  - RMSE: 0.0087

  - R^2 Score: 0.9998

Step 4: Advanced Modeling

- Objective: Improve model performance using advanced techniques.
- Process:

  - Feature Selection:

    - Lasso Regularization:

    - Selected important features using Lasso Regularization.

    - Important Features: ['segment_actual_time.1', 'segment_actual_time trip_creation_month']

  - Recursive Feature Elimination (RFE):

  - Selected important features using RFE with Random Forest Regressor.

- Model Training:

  - Artificial Neural Network (ANN):

    - Trained an ANN model with 50 epochs.

    - Evaluated the model using MSE, RMSE, and R^2.

  - Hyperparameter Tuning:

    - Used Randomized Search for XGBoost and LightGBM models.

    - Best parameters and scores were identified for both models.

- Output:

  - Lasso:

    - Important Features: segment_actual_time and segment_actual_time
    - Mean Squared Error: 5.963439031210556e-05
    - Root Mean Squared Error: 0.007722330626961369
    - Mean Absolute Error: 0.006733926616145296
    - R^2 Score: 0.9998036200052525

  - ANN Evaluation Matrix:

    - MSE: 96.55888768720551
    - RMSE: 9.826438199429411
    - R^2: -316.97480879074936

  - Best parameters for XGBoost: {'subsample': 0.8, 'n_estimators': 100, 'max_depth': 7, 'learning_rate': 0.1, 'colsample_bytree': 0.8

  - Best parameters for LightGBM: {'num_leaves': 31, 'n_estimators': 200, 'max_depth': -1, 'learning_rate': 0.1}

Step 5: Cross-Validation

- Objective: Ensure model robustness using K-Fold Cross-Validation.
- Process:

  - Used 5-fold cross-validation to evaluate model performance.

  - Documented cross-validation scores and saved them as a CSV file.

- Output:
  - Mean CV MAE: 0.0004
  - Standard Deviation of CV MAE: 0.0007

- Mean CV R^2: 0.9999

Step 6: Ensemble Methods

- Objective: Combine multiple models to improve predictive performance.
- Process:

  - Model Averaging:

  - Averaged predictions from Linear Regression, Ridge Regression, and Random Forest models.

  - Evaluated averaged ensemble predictions.

  - Stacking:

    - Used a Stacking Regressor with Linear Regression as the meta-model.

    - Evaluated stacking ensemble predictions.

  - Comparison:

    - Compared results from individual models, model averaging, and stacking.

- Output:

  - Model Averaging MAE: MAE: 0.0091, RMSE: 0.0143, R^2: 0.9993

  - Stacking MAE: MAE: 0.0021, RMSE: 0.0087, R^2: 0.9998

  - Comparison of Models:

    - Linear Regression MAE: 0.0019
    - Ridge Regression MAE: 0.0145
    - Random Forest MAE: 0.0184
    - Model Averaging MAE: 0.0091
    - Stacking MAE: 0.0021

Step 7: Performance Documentation

Objective: Summarize Model Performance Metrics and visualize feature importance

Process:

- visualize ensemble results, feature importance, residuals, and performance metrics and learning curve.

- Save all generated metrics and plots.

- Output:

  - Visualization of all above

Step 8: Deployment Preparation

- Objective: Save Best Model and Prepare the model for deployment
- Process:
  - Save Best Model
  - Created an inference pipeline to load the saved model and make predictions.
  - Provided example usage of the inference function.
- Output:
  - Best model saved as 'best_model.pkl'.
  - make_prediction function created and tested with sample input.

Step 9: Model Usage Documentation

- Objective: provide a comprehensive guide on how to use the trained model (best_model.pkl) for making predictions on new data.
- Process:
  1- Load the Model
  2- Input Data
  3- Make Prepare prediction
- Output:
  - NumPy array of floats

Conclusion

The steps involved integrating multiple datasets, performing feature selection, training various models, and using ensemble techniques to improve predictive performance. The final model, a Stacking Regressor, demonstrated superior performance compared to individual models and model averaging. The model was saved and prepared for deployment, ensuring it can be used for future predictions. The entire process was documented, and all results were saved for reference.