Bike Trip Duration Prediction

Linlin Yu   03/2025

The goal of this project is to develop a machine learning model to predict bike trip durations based on various features including station locations, weather conditions, day and time, and user demographics. In addition to the attached **experiment jupyter notebook**, See also:

- **Project Code Repository:**
  https://github.com/Franklin112233/Bike-Task

- **Interactive Demo App:**
  http://172.237.108.162:8080/

1. Data Exploration

The database was created by combining three provided CSV files containing information about stations, trips, and weather.

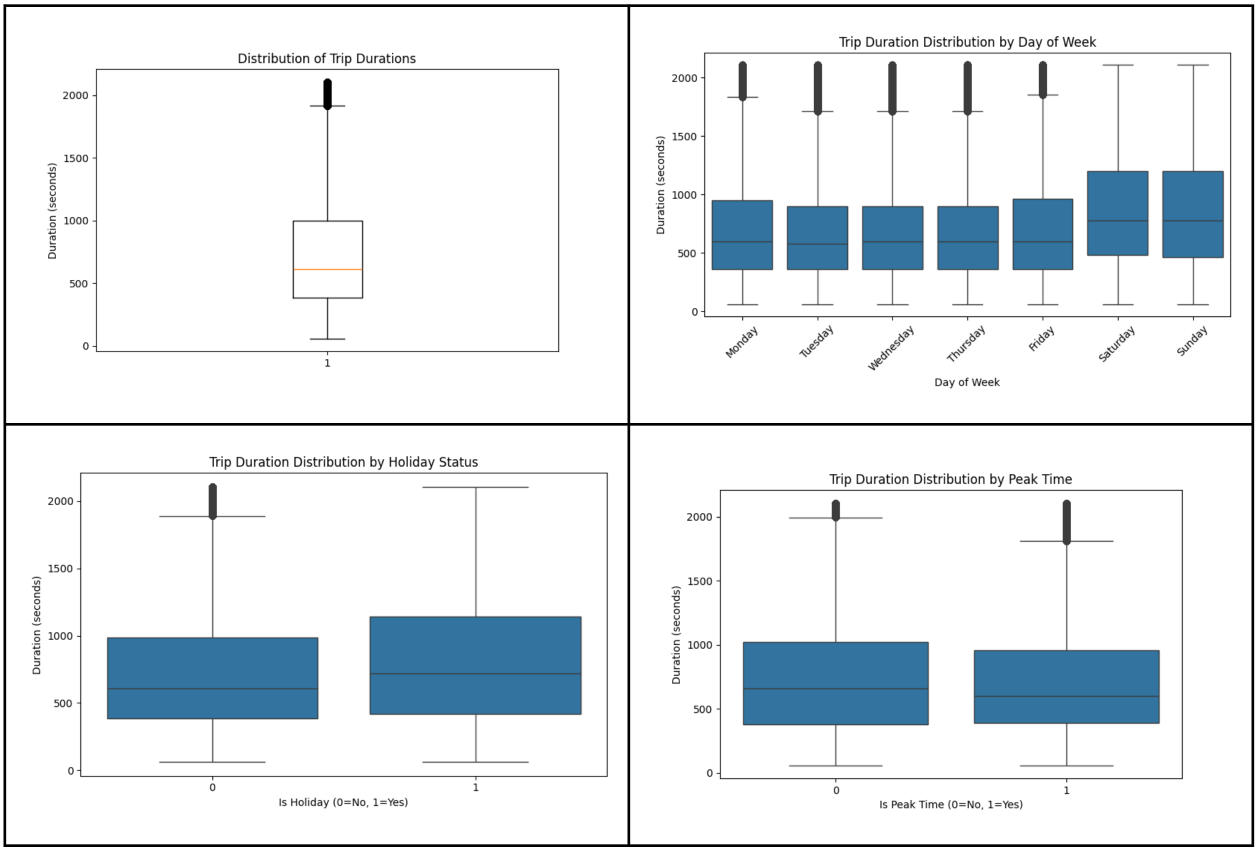
An analysis of the trip duration distribution revealed the presence of outliers, which could significantly skew model performance. To mitigate this issue, The duration data was constrained to include only trips lasting between 60 seconds and 24 hours, establishing a reasonable range for typical bike-sharing journeys. Following the initial filtering, a specialized outlier removal function was applied to give a further outlier removal.

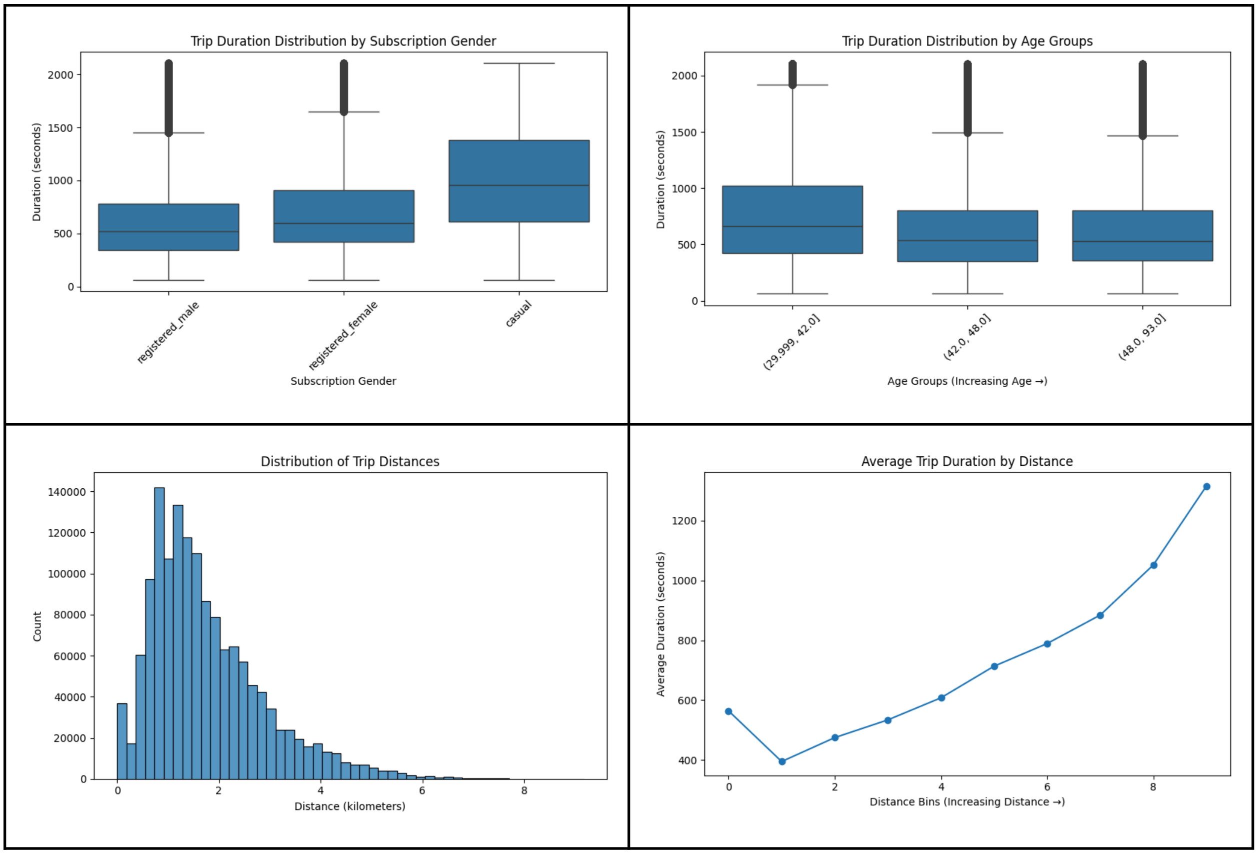
Traffic patterns can vary significantly depending on holidays, weekends, and the time of day (e.g., peak vs. off-peak hours). To capture these variations, relevant features were extracted from the start_date column.

User demographics were also found to play a significant role. Differences in behavior were observed across gender (male vs. female), age groups, and user types (registered vs. casual users). These factors were created and added as features into the dataset.

Trip duration is strongly influenced by the distance traveled. Using the provided coordinates for start_station and end_station, a distance function was implemented to calculate the distance between two stations, which was added as a key feature in the dataset.

HPCP data is available from the weather dataset; however, it contains a significant proportion of missing values. To address this issue, an imputation mechanism was implemented. When an HPCP value is missing, it is replaced with the most recent hour's HPCP value. This approach ensures continuity in the data while minimizing the impact of missing values on subsequent analyses.





2. Model Training and Evaluation

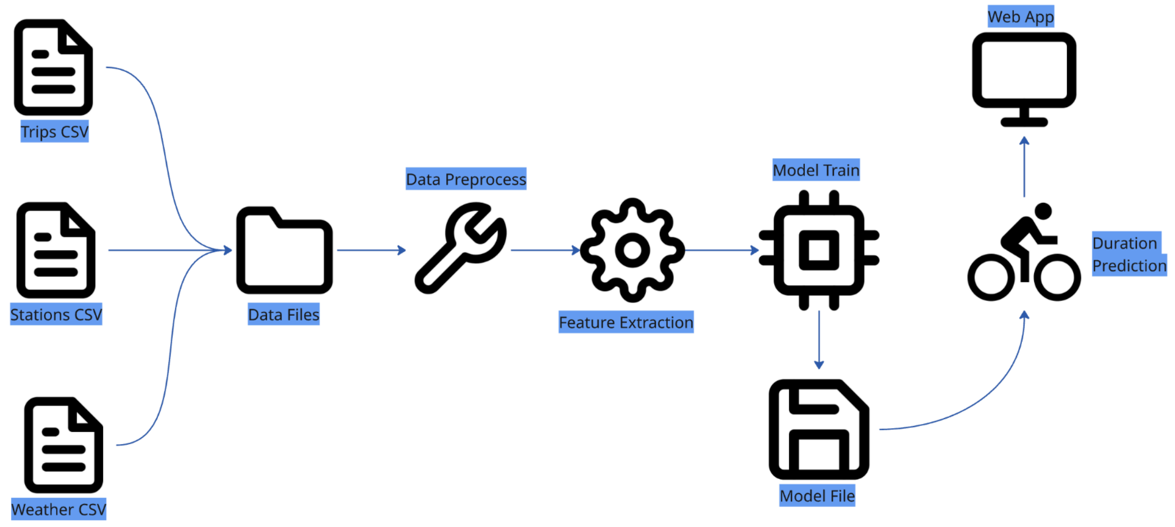
A comprehensive machine learning pipeline was developed to predict the target variable of trip "Duration".The data was partitioned into 75% training and 25% testing sets. A robust preprocessor was implemented, standardizing numerical features and one-hot encoding categorical variables. The model pipeline integrated this preprocessor with three distinct regressors: Linear Regression, Random Forest, and XGBoost. For each regressor, specific hyperparameter search spaces were defined to facilitate tuning. A Bayesian optimization strategy with cross-validation was employed for automated hyperparameter selection, aiming to minimize prediction errors. Model performance was evaluated using two key metrics: R-squared (R^2) to quantify explained variance and Root Mean Squared Error (RMSE) to measure prediction accuracy in the original units of duration.

Model performance metrics are shown as below: according to the result, Random Forest Model with 300 estimators and depth of 5, has a best R2 score and lowest

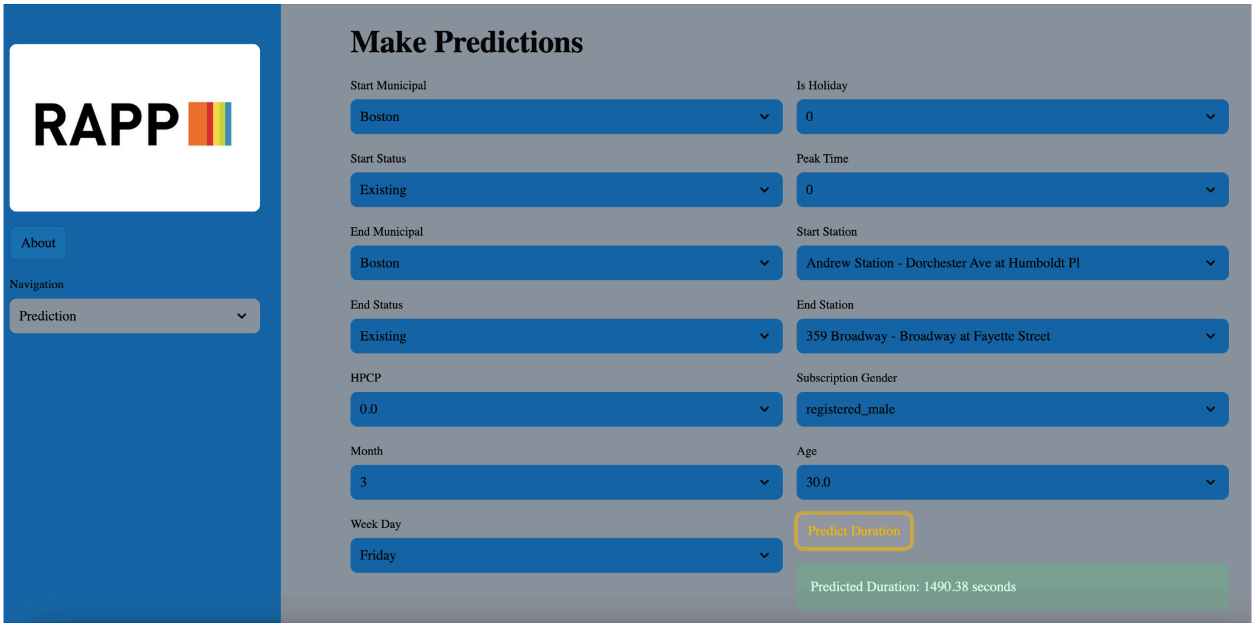
RMSE among all models. It has been selected as the best model for the prediction purpose.

Linear Regression	Random Forest	XGBoost
Best score: 0.541 Best parameters: OrderedDict({'regressor__fit_intercept': True}) Test R2 Score: 0.474 Test RMSE: 0.441	Best score: 0.643 Best parameters: OrderedDict({'regressor__max_depth': 5, 'regressor__n_estimators': 300}) Test R2 Score: 0.659 Test RMSE: 0.355	Best score: 0.626 Best parameters: OrderedDict({'regressor__max_depth': 3, 'regressor__n_estimators': 101}) Test R2 Score: 0.620 Test RMSE: 0.375

3. Prediction and Deployment



The project follows a workflow encompassing data preparation, data exploration, feature engineering, model training, and model development. Initially, CSV tables are merged to create consolidated data files, which then undergo preprocessing and cleaning. Feature engineering is applied to generate locational, demographic, temporal, and weather-related attributes crucial for model training. A machine learning pipeline is constructed to train the model, with the best-performing model and its associated performance metrics saved as artifacts. For prediction, this optimized model is retrieved and utilized to forecast based on live data inputs. To showcase the project and facilitate user interaction, a frontend application is developed, allowing users to input live data for predictions. The entire system, including the prediction service and web application, is containerized using Docker and deployed on a remote virtual server, ensuring scalability and ease of maintenance.



Navigation: Prediction

Make Predictions

Start Municipal: Boston

Start Status: Existing

End Municipal: Boston

End Status: Existing

HPCP: 0.0

Month: 3

Week Day: Friday

Is Holiday: 0

Peak Time: 0

Start Station: Andrew Station - Dorchester Ave at Humboldt Pl

End Station: 359 Broadway - Broadway at Fayette Street

Subscription Gender: registered_male

Age: 30.0

Predicted Duration: 1490.38 seconds

4. Further Improvement

As this project was developed as part of an interview task, data preprocessing and model training were conducted on a local machine due to time and computational resource constraints. While the current implementation is functional, several areas offer potential for future improvement.

First, only a small dataset and three types of models were used to optimize training speed. Expanding the dataset and exploring additional models could improve performance, especially if cloud computing resources are utilized to overcome local hardware limitations.

Similarly, incorporating more detailed weather information, when available, could enhance the model's predictive capabilities. The current weather column did not contribute significantly to the model, suggesting the need for further refinement or feature engineering.

Additionally, while distance is a key factor in predicting trip duration, future iterations could consider integrating population density and traffic patterns at each location to provide a more comprehensive view.

Time-based trends also appear relevant but did not demonstrate strong evidence in the current model. A deeper analysis of temporal data—such as extracting advanced time-series features using libraries like tsfresh—could uncover valuable insights for improving predictions.

Finally, employing recursive feature elimination (RFE) during model training could help identify the optimal selection of features for better performance. It would benefit from cloud-based resources to handle the increased workload efficiently.