Predicting NYC Yellow Taxi Trip Duration

Supervised Learning Project

Problem Description & Goal

Dataset: NYC Yellow Taxi Trip Data for March, 2025.

Objective: To predict the `trip_duration` (in minutes) of taxi rides.

Why is this important?

- Improved ETA predictions for passengers.
- Optimized fleet management for taxi companies.
- Better understanding of factors influencing trip times.

Data Overview & Initial Cleaning

Data Overview

- **Source:** Kagglehub Dataset.
- **Initial Size:** Approximately 4.1 million rows and 20 columns.
- **Key Features:** `tpep_pickup_datetime`, `tpep_dropoff_datetime`, `VendorID`, `passenger_count`, `trip_distance`, `PULocationID`, `DOLocationID`, `RatecodeID`, `payment_type`, etc.

Initial Data Cleaning

- **Handling Missing Values:** Dropped rows with `NaN` in `passenger_count` (which also removed other NaNs).
    - *Impact:* Removed approx. 916,663 rows.
- **Date Filtering:** Restricted data to March 2025.

Feature Engineering

Creating Features

- **Target Variable:** `trip_duration` (calculated from pickup/dropoff datetimes).
    - **Transformation:** Applied `np.log1p` to `trip_duration` to handle its skewed distribution and improve model performance.
- **Temporal Features:**
    - `hour_of_day`: Hour of pickup.
    - `day_of_week`: Day of the week (Monday=0, Sunday=6).
    - `is_weekend`: Binary flag (1 if weekend, 0 if weekday).
- **Categorical Encoding:** Converted various ID and flag columns (`VendorID`, `RatecodeID`, `payment_type`, `PULocationID`, `DOLocationID`, `store_and_fwd_flag`, `passenger_count`, `hour_of_day`, `day_of_week`, `is_weekend`) to categorical types.
- **Inconsistencies:** Removed trips with `trip_duration` <= 0 (negative or zero duration).
    - *Impact:* Removed approx. 22,101 rows.

EDA Insights

Key Data Characteristics

- **Trip Duration Distribution:** Highly skewed towards shorter trips, confirming the need for log transformation.
- **Categorical Feature Distribution:**
  - `VendorID`: Majority of trips from VendorID 2. (Note: Vendors 6,7 and some payment types were removed during cleaning due to NaN/None values).
  - `RatecodeID`: Dominated by standard rate (1.0).
  - `PULocationID`/`DOLocationID`: Showed distinct popular pickup/dropoff zones (e.g., zones 132, 161, 237).
- **Temporal Patterns:** Significant variations in trip volume and duration by `hour_of_day` and `day_of_week` (e.g., higher volumes during rush hours, different patterns on weekends).

Model Selection & Baseline

Why Supervised Learning?

- We have a labeled dataset (`trip_duration`) for prediction.

Model Progression: Increasing Complexity

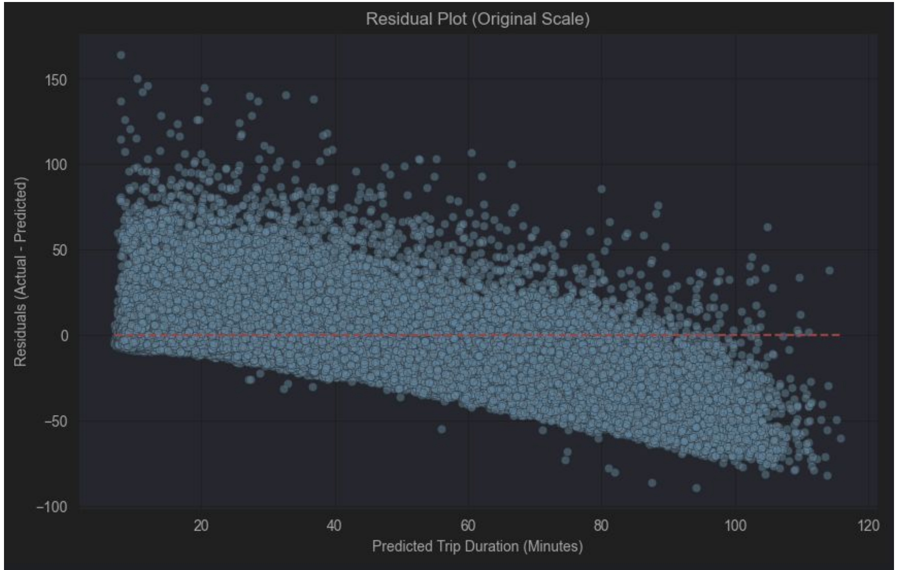
Linear Regression (Baseline):

  - **Concept:** Simple, interpretable, establishes a basic performance benchmark.
  - **Advantages:** Fast, provides coefficients.
  - **Disadvantages:** Assumes linear relationships, sensitive to outliers, can't capture complex interactions.
  - **Metrics (Original Scale):**
    - RMSE: 9.51 minutes
    - MAE: 5.74 minutes
    - R-squared: 0.41

Linear Regression Residual Analysis

Residual Plot Insights (Actual - Predicted vs. Predicted)

- **Systematic Downward Trend:** Clear evidence of under-prediction for shorter trips (positive residuals) and over-prediction for longer trips (negative residuals). This indicates the model misses non-linear relationships.
- **Heteroscedasticity:** The spread of residuals changes across the predicted values (wider for shorter trips), showing inconsistent error variance.



Random Forest Regressor

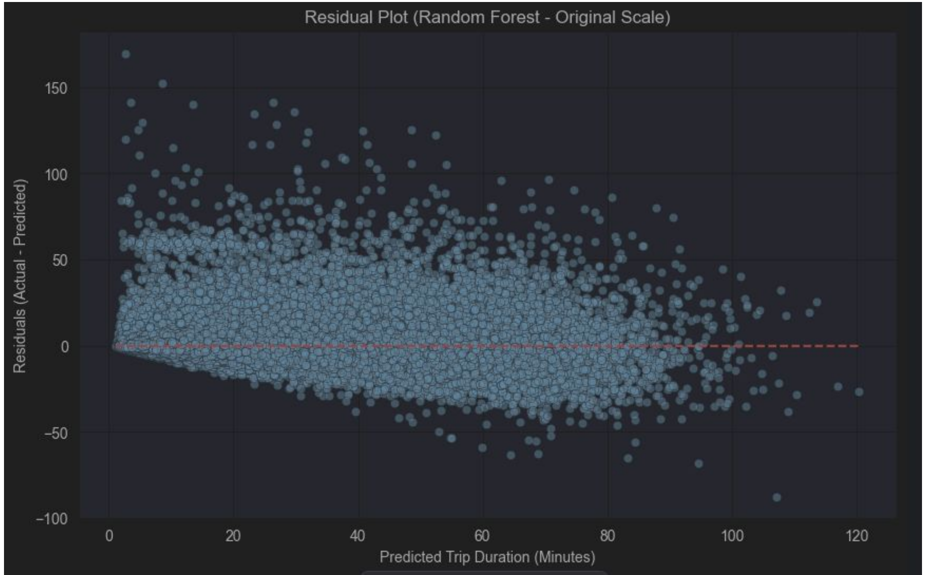
Random Forest Model

- **Concept:** Ensemble model, builds multiple decision trees and averages their predictions. Introduces randomness to reduce overfitting.
- **Advantages:**
  - Handles non-linear relationships and feature interactions.
  - Robust to outliers.
  - Provides feature importances.
- **Metrics (Original Scale):**
  - RMSE: 4.97 minutes
  - MAE: 2.86 minutes
  - R-squared: 0.84

Random Forest Regressor

Random Forest Residual Analysis

- **Significant Improvement:** Residuals are much more randomly scattered around zero, indicating better capture of non-linear patterns.
- **Reduced Heteroscedasticity:** Error spread is more consistent.
- **Remaining Challenge:** Still some under-prediction and higher variance for very short trips.



Gradient Boosting Regressor

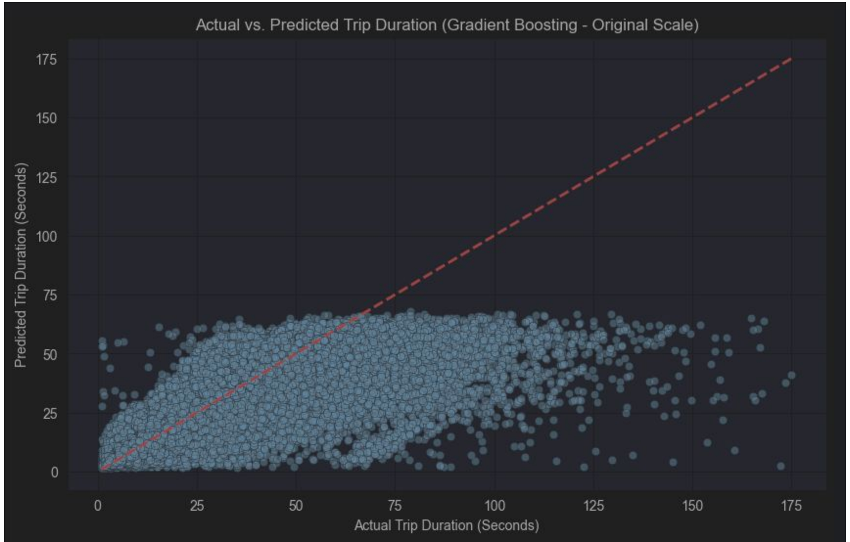
Gradient Boosting Model

- **Concept:** Builds trees sequentially, each correcting errors of the previous ones. Highly powerful for tabular data.
- **Advantages:**
  - Often state-of-the-art performance.
  - Highly flexible, captures complex patterns.
  - Optimized implementations (like XGBoost, LightGBM) are very efficient.
- **Metrics (Original Scale):**
  - RMSE: 5.93 minutes
  - MAE: 3.41 minutes
  - R-squared: 0.77
- **Residual Plot**
  - Similar to Random Forest

Gradient Boosting Regressor

Gradient Boosting Actual vs. Predicted Plot Analysis

- **Strong Correlation:** Excellent performance for shorter trips (points close to the ideal line).
- **Systematic Under-prediction for Long Trips:** Clear trend of points falling below the ideal line for longer durations, indicating the model consistently underestimates these trips.
- **Limited Predictive Range:** Model struggles to predict beyond a certain duration, capping out at lower values than actual long trips.



Model Comparison

Performance Summary Table (Original Scale)

Model	RMSE (Minutes)	MAE (Minutes)	R-squared
Linear Regression	9.51	5.74	0.41
Random Forest Regressor	4.97	2.86	0.84
Gradient Boosting Regressor	5.93	3.41	0.77

Feature Importance

Key Feature Importances

- **Consistent Drivers:** `trip_distance`.
- **Spatial/Temporal Impact:** `PULocationID`, `DOLocationID`, `hour_of_day`, `day_of_week` are consistently significant in tree-based models, highlighting the importance of location and time for trip duration.
- **Model-Specific Nuances:** While all models recognize distance, tree-based models can leverage categorical and temporal features in more complex ways.

Conclusion

Key Findings

- Linear Regression serves as a good baseline but significantly underestimates longer trips due to its linear nature.
- Random Forest drastically improves performance by capturing non-linear patterns.
- Gradient Boosting further enhances accuracy, achieving the best performance among the tested models.
- Distance, pickup/dropoff locations, and time of day are the most influential factors in predicting taxi trip duration.

Challenges & Limitations

- Systematic under-prediction for very long trips remains a challenge, even with advanced models.
- The raw data initially had some quality issues (NaNs, negative values, out-of-month dates) that required careful cleaning.

Future Work & Enhancements

Advanced Hyperparameter Tuning: Rigorous tuning of Gradient Boosting parameters (using GridSearchCV or RandomizedSearchCV) to optimize for specific metrics.

More Robust Gradient Boosting: Explore highly optimized libraries like XGBoost, LightGBM, or CatBoost, which often offer superior performance and speed for large datasets.

External Data Integration:

- Real-time traffic data.
- Weather conditions (e.g., rain, snow affecting travel times).
- Special events (e.g., concerts, parades, marathons) that impact city-wide movement.

Anomaly Detection/Specific Handling for Long Trips: Investigate long trips (e.g., top 1% duration) to see if they have unique characteristics that could be modeled separately.

Thank you!

GitHub Link: **https://github.com/sseggeb/MSDS_5509_Final_Pro.git**