NYC Taxi Trip Duration Prediction

A Machine Learning Approach Focused on Feature Engineering and Model Optimization

Project Overview

This project aims to predict the duration of taxi trips in New York City using machine learning techniques, with a strong focus on effective feature engineering and model evaluation. It is structured into two main parts:

- **Part I: Feature Engineering and Baseline Modeling**
  This part concentrates on extensive exploratory data analysis and crafting interpretable features. A Ridge Regression model with fixed regularization ($\alpha = 1$) is used to evaluate the isolated impact of these engineered features on prediction accuracy.
- **Part II: Model Comparison and Hyperparameter Tuning**
  Building upon the foundation from Part I, this phase explores different machine learning models, including Neural Networks, Ridge Regression, XGBoost, and a stacked ensemble combining Ridge and XGBoost as base learners with a Neural Network as the meta-learner. Hyperparameter tuning is performed using randomized search to optimize each model's performance. The models will then be compared against each other to select the best model for production deployment.

Taher Alabbar

July 31, 2025

Project Workflow Overview

This diagram summarizes the key steps of the taxi trip duration prediction project, from data preparation through to model deployment.

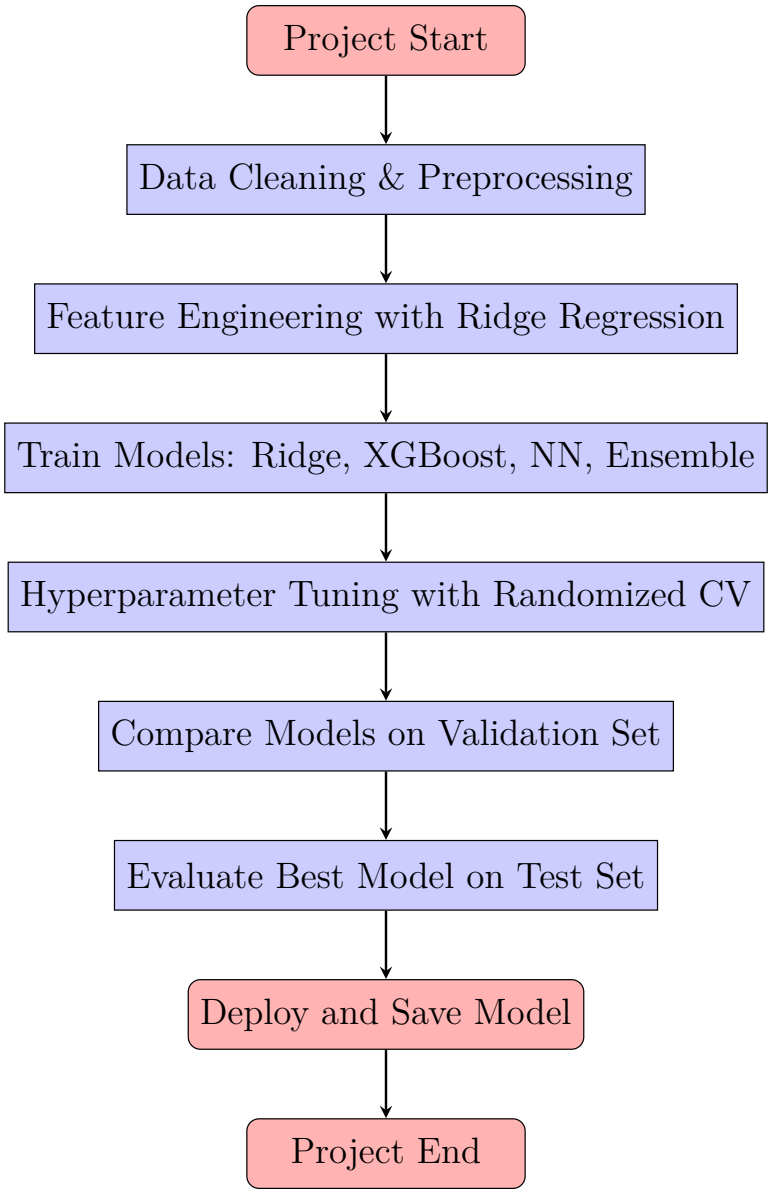

Figure 1: Project workflow: From data processing to model deployment.

Part I: Feature Engineering and Baseline Modeling

This part is primarily dedicated to feature engineering and evaluating its impact on model performance. The goal is to extract informative patterns from the raw dataset through thoughtful transformations and domain-driven insights. To ensure that improvements are solely due to feature design and not model complexity, all experiments in this part are conducted using Ridge Regression with fixed regularization ($\alpha = 1$).

1. Feature Descriptions

The dataset includes the following features:

- `id`: Unique identifier for each trip.
- `vendor_id`: Code indicating the vendor providing the trip.
- `pickup_datetime`: Timestamp when the trip started.
- `passenger_count`: Number of passengers in the taxi.
- `pickup_longitude`: Longitude where the trip started.
- `pickup_latitude`: Latitude where the trip started.
- `dropoff_longitude`: Longitude where the trip ended.
- `dropoff_latitude`: Latitude where the trip ended.
- `store_and_fwd_flag`: Indicates whether the trip data was stored before forwarding to the server.
- `trip_duration`: Total trip time in seconds (target variable).

2. Insights from the Data

The following insights were derived from an exploratory data analysis (EDA) of the dataset. They guided preprocessing decisions and informed feature engineering:

Note: Temporal features such as ***`hour`****,* ***`minute`****,* ***`day`****,* ***`month`****, and* ***`weekday`*** *were extracted from the* ***`pickup_datetime`*** *field to enable detailed time-based analysis.*

- **trip_duration**: The target variable displayed a heavy skew with many extremely short trips (around 1 second), likely due to data errors or very short rides, and extreme outliers lasting up to 40.8 days, which are unrealistic for taxi trips. These outliers were removed to improve data quality. Applying a log transformation stabilized variance, reduced skewness, and enhanced correlations with predictor variables, which improved the robustness and interpretability of the model.

- **passenger_count**: Values of 0, 7, and 8 were identified as outliers—either implausible (0 passengers) or exceeding typical taxi capacity—and were removed to maintain data integrity. Most trips involved a single passenger, reflecting standard taxi usage. Trips with higher passenger counts often correspond to group travel or airport trips, which tend to be longer and may follow different spatial or temporal patterns.

- **month**: Seasonal trends were evident, with trip durations longest in *summer* months, likely due to vacation travel and increased leisure activity, followed by *spring*. The shortest durations occurred in *winter*, which may relate to reduced travel or weather constraints impacting trip behavior.

- **hour**: Time of day strongly affected trip duration. Trips during the *afternoon* generally lasted the longest, corresponding to peak traffic and rush hour congestion. Evening, morning, and night periods followed in decreasing order of trip duration, highlighting the influence of daily traffic cycles on travel time.

- **weekday**: Trip durations increased progressively from *Monday to Thursday*, reflecting typical workweek traffic buildup, and then decreased towards the weekend, with *Sunday* having the shortest durations. This pattern suggests weekday commuting significantly influences trip length and traffic conditions.

- **vendor_id**: Although vendor usage was balanced, trips associated with `vendor_id = 2` tended to have slightly longer durations on average. This could reflect differences in geographic service areas, routing strategies, or customer base between vendors.

- **store_and_fwd_flag**: This feature indicates whether trip data was temporarily stored in the vehicle before being sent to servers, usually due to connectivity issues. The vast majority of trips had a flag of `'N'`, meaning real-time transmission. Trips flagged `'Y'` were rare but typically had longer durations, possibly reflecting rural or low-connectivity areas. Due to the significant class imbalance and minimal predictive value, this feature was dropped.

- **pickup_latitude and dropoff_latitude**: These coordinates individually showed many rare, possibly erroneous values. Combining them into a single feature (`latitude_sum`) provided a more stable spatial representation that facilitated outlier detection and improved model interpretability related to geographic patterns.

- **pickup_longitude and dropoff_longitude**: Similar to latitude, these features were combined into `longitude_sum` to better capture spatial location and clustering. This helped manage geographic variability and improve the model's ability to learn location-related effects on trip duration.

- **Geographic Insight (from lat/lon)**:

  - *Northwest Region*: Represents dense urban zones such as Midtown and Downtown Manhattan characterized by a grid street system and heavy taxi activity, which influences trip patterns and durations.
  - *Southeast Region*: Corresponds to airport areas such as JFK, where trips often follow longer, curved routes, typically resulting in longer durations.

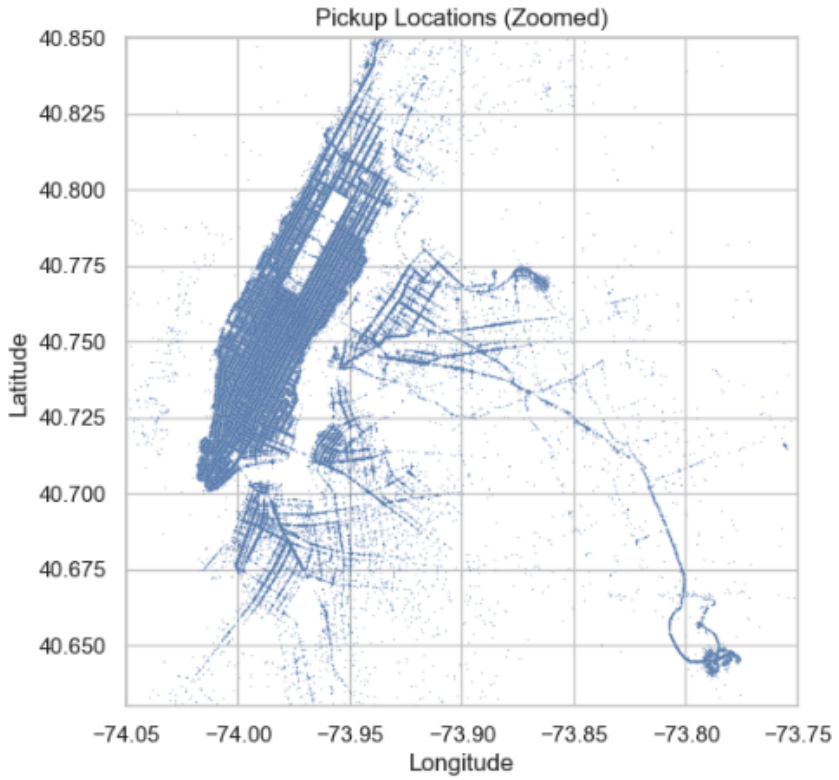

Figure 2: Pickup & Dropoff Density Plot showing spatial concentration of taxi activity.

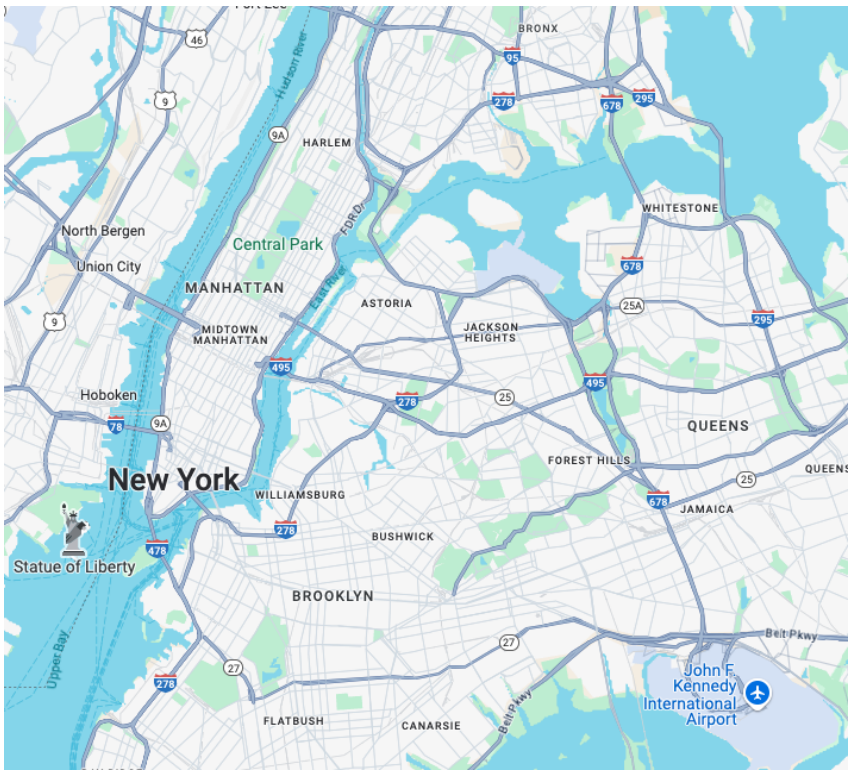

Figure 3: Reference Map of NYC highlighting key regions (e.g., Manhattan, JFK Airport) used for comparison.

3. Data Cleaning & Preprocessing

Outlier Detection with IQR

- Applied the Interquartile Range (IQR) method to remove extreme outliers from key features such as:
  - pickup_latitude, pickup_longitude (Dropped post-feature engineering)
  - dropoff_latitude, dropoff_longitude (Dropped post-feature engineering)
  - trip_duration (after log transform)

  The Interquartile Range (IQR) is defined as:

$$\text{IQR} = Q_3 - Q_1$$

  where Q_1 and Q_3 are the 25th and 75th percentiles respectively.

  Outliers are typically identified as data points lying outside the range:

$$[Q_1 - k \times \text{IQR}, \quad Q_3 + k \times \text{IQR}]$$

  where k is the multiplier controlling the strictness of outlier removal.

- Experimented with different multipliers (denoted as k) to control strictness:

IQR Multiplier k	R^2 Score (Small Dataset)
1.5	0.60221
5	0.65984
8 (Chosen)	**0.68638**
10	0.68608

- Settled on $k = 8$ for best balance between outlier removal and data retention.
- Observed that aggressive filtering (e.g., $k = 1.5$) removed legitimate long trips, harming model generalization.

Log Transformation of Target

Applied:

$$\text{log\_trip\_duration} = \log(\text{trip\_duration} + 1)$$

Benefits:

- Reduced right skew
- Stabilized variance
- Increased feature correlation
- Enabled more stable model training

Datetime Parsing

Extracted temporal features directly from the `pickup_datetime` column:

- `hour`: Hour of the trip (0–23)
- `minute`: Minute of the trip (0–59)
- `day`: Day of the month
- `month`: Month of the year
- `weekday`: Day of the week (0 = Monday, 6 = Sunday)

Feature Selection and Encoding

- Dropped columns `id`, `pickup_datetime`, `pickup_latitude`, `dropoff_latitude`, `dropoff_longitude`, `pickup_longitude`, and `store_and_fwd_flag` after feature engineering.
- One-hot encoded categorical features including `month`, `hour`, `passenger_count`, `weekday`, `minute`, `weekend_times_month`, `hour_times_weekend`, `month_times_weekend`, and `vendor_passenger_interaction`.
- StandardScaler applied to all remaining numeric features.

Note on IQR-Based Outlier Removal

To prevent data leakage and ensure realistic preprocessing, IQR thresholds for outlier detection were calculated **exclusively from the training set** and then consistently applied to the validation and test sets. This approach simulates real-world deployment, where new data is filtered using fixed boundaries derived from historical data, ensuring consistency and reproducibility without peeking into future information.

4. Engineered Features Used

The following features were engineered and selected based on their interpretability and positive contribution to model performance:

- **1. Spatial Features (Distance and Location):**
  - `trip_distance`: Haversine distance between pickup and dropoff points; the most informative spatial feature.
  - `trip_distance_squared`: Captures nonlinear effects of distance on trip duration.
  - `log_trip_distance`: Reduces skewness, stabilizes variance, and enhances model interpretability.
  - `latitude_sum`, `longitude_sum`: Approximate location of trip to capture neighborhood-level effects.

- `distance_times_latitude_sum`, `distance_times_longitude_sum`: Interaction between trip area and trip length.

Note: Distance and its variants significantly improved model performance. While multiple forms may induce multicollinearity, empirical testing showed performance gains. We also observed that short trips often had disproportionately long durations due to urban traffic, while longer trips exhibited zigzag patterns in duration due to road and route variability.

- **2. Temporal Features (Time-Based Context):**
  - `is_weekend`, `is_night`, `is_morning`, `is_afternoon`, `is_evening`: Time-of-day and day-of-week indicators for traffic and demand patterns.
  - `is_summer`, `is_spring`, `is_winter`: Seasonal flags reflecting variations in weather, traffic, and behavior across months.

- **3. Interaction Features (Compound Effects):**
  - `night_and_weekend`, `weekend_times_month`, `hour_x_is_night`, `hour_times_weekend`, `month_times_weekend`: Capture overlapping effects in time (e.g., nighttime on weekends).
  - `hour_x_passenger_count`, `trip_distance_x_passenger_count`, `trip_distance_x_weekday`, `vendor_passenger_interaction`: Reflect behavioral and operational dynamics, such as passenger load at different times or vendor-specific trip patterns.

Note: Not all engineered features were used in the final model. The above list includes the selected engineered features settled upon for model training and inference.

Additionally, the analysis of many engineered features shown below was performed during training to evaluate their individual impact on model performance. The feature list and evaluation samples presented here represent only a subset of the full feature assessment process carried out to guide final feature selection.

Spatial Features

Feature	Outcome
trip_distance (Haversine)	Most impactful single feature
trip_distance_squared	Captured non-linearity, improved R^2
log_trip_distance	Helped reduce skew
latitude_sum, longitude_sum	Outperformed raw lat/lon individually
lat_diff, lon_diff	Less informative compared to sum features
lat_sum_x_lon_sum, *_squared	No added value
distance_x_lat_sum, distance_x_lon_sum	Provided strong improvements

Temporal Features

Feature	Effect
hour_x_is_night	Captured peak-hour night behaviour
month_x_weekend, hour_x_weekend	Helpful for seasonal/cyclical interaction
is_rush_day, rush_weekday	Degraded performance — likely redundant info
distance_per_day	Very slight improvement

Categorical / Behavioral Features

Feature	Insight
passenger_count (OHE)	High counts (7, 8) removed as outliers
store_and_fwd_flag	Dropped — heavily imbalanced and added no value
vendor_id_x_passenger_count	Surprisingly strong — improved R^2 by ~ 0.01
trip_distance_x_passenger_count	Captured group travel trends
trip_distance_x_weekday	Captured weekday vs. weekend travel distance shifts
one_passenger flag	Useless — already dominant case, added no signal

Key note: Even when engineered features are individually informative and distinct, using all of them simultaneously as separate input features does not always improve model performance. Sometimes, selecting a subset yields better results, as the model may struggle to learn patterns from all features due to noise, redundancy, or conflicting information. This refers to including multiple features together in the model, not merging them into a single feature.

5. Modeling Approach

Component	Description
Model	Ridge Regression ($\alpha = 1$)
Preprocessing	StandardScaler for numeric, OneHotEncoder for categorical
Validation	Simple train/validation split using consistent IQR thresholds
Evaluation	R^2 Score and RMSE on log-transformed target

6. Final Performance

Metric	Training Set	Validation Set
R^2 Score	0.68638	0.68688
RMSE	0.43218	0.43290

7. Key Insights

- Trip distance and its transformations are the core drivers of trip duration.
- Lat/Lon sums are more useful than their differences — possibly because they reflect absolute position and area (e.g., Manhattan vs. airport), which affects expected trip length.
- Outlier calibration ($k = 8$) had a more significant effect on model performance than many features.
- Over-engineering or redundant flags (e.g., rush indicators, one-passenger flag) harmed model performance.
- Empirical testing guided all decisions — correlation alone was not enough.
- The most impactful features were those that:
    * Reflected **real-world relationships**, such as spatial layout, trip length, and temporal dynamics.
    * Applied **mathematically appropriate transformations** to address skewness or non-linearity.
    * Were **evaluated through model performance**, not just intuition.

Conclusion
This project demonstrates **the critical role of careful feature engineering** in enhancing model performance — especially when using a constrained model like Ridge Regression. By focusing on well-reasoned transformations, spatial/temporal interactions, and principled outlier handling, we achieved **robust performance with interpretable features**.

Part II: Model Comparison and Hyperparameter Tuning

Building on the feature engineering from Part I, this phase evaluates and optimizes several models for predicting taxi trip durations, including Ridge Regression, XGBoost, a Neural Network, and a Stacked Ensemble combining RR and XGB with a Neural Network meta-learner.

Hyperparameter Tuning

Each model underwent hyperparameter tuning using Randomized Search with cross-validation to identify optimal configurations. This approach allowed efficient exploration of a broad hyperparameter space, enhancing model generalization and performance. The final hyperparameters for the selected XGBoost model were:

- `max_depth = 13`
- `n_estimators = 137`

Model Performance Comparison

All models were compared based on their performance on the validation set to select the best-performing one. The table summarizes the results:

Model	Train R^2	Validation R^2	Train RMSE	Validation RMSE
XGBoost (XGB)	0.856	**0.759**	0.293	**0.381**
Stacked Model	0.777	0.754	0.365	0.385
Neural Network (NN)	0.751	0.747	0.385	0.390
Ridge Regression (RR)	0.686	0.689	0.432	0.433

Table 1: Model performance comparison on training and validation sets.

Final Model Evaluation on Test Set

Based on the superior validation performance, XGBoost was selected as the final model. It was retrained on the full training (train + val) dataset using the best hyperparameters and then evaluated on the held-out test set to assess its generalization capability. The final performance is shown below:

Dataset	R^2	RMSE
Training Set	0.847	0.302
Test Set	0.759	0.380

Table 2: Final performance of the selected XGBoost model on the test set.

Conclusion

Through comprehensive model evaluation and hyperparameter tuning, XGBoost consistently emerged as the top-performing model. It achieved the highest validation R^2 of 0.759 and the lowest RMSE of 0.381, demonstrating strong predictive performance on unseen data. This trend held during final testing, where XGBoost achieved a test R^2 of 0.759 and RMSE of 0.380, confirming its robust generalization capabilities.

While the stacked ensemble model and Neural Network also delivered competitive results, their added complexity did not yield significant gains over XGBoost. Ridge Regression, serving as a linear baseline, lagged behind in both R^2 and RMSE metrics.

Given its superior performance, efficiency, and ease of deployment, XGBoost is selected as the final model for production. Its ability to handle nonlinearity, robustness to outliers, and interpretability through feature importance make it a compelling choice for real-world applications in taxi trip duration prediction.