Report: End-to-End Machine Learning on NYC Taxi Trip Duration Dataset

Author: Kainat Khalid
Course: Machine Learning (SP2025)
Instructor: Basharat Hussain
Date: 16th-Feb-2025

📜 Table of Contents

1️⃣ Introduction

Machine Learning models require **structured preprocessing**, **training**, and **evaluation** to make accurate predictions. This project explores an **end-to-end ML workflow** on the **NYC Taxi Trip Duration Dataset**, aiming to predict **trip duration** based on multiple features.

- **Domain:** Transportation
- **Problem Type:** Regression
- **Dataset Size:** 1M+ records
- **Target Variable:** `tripDuration` (continuous variable)

We follow a structured approach:

1. **Understand the dataset & perform EDA 🕵️‍♂️**
2. **Handle missing values & categorical attributes 🛠️**
3. **Train multiple models & evaluate performance 📊**
4. **Optimize the best model using hyperparameter tuning 🎯**
5. **Deploy and interpret the model for better insights 🎹**

---

2 Dataset Selection & Understanding

The dataset was selected from **Kaggle**:
NYC Taxi Trip Duration Dataset.

📌 Key Dataset Details

Feature Name	Description
`passengerCount`	Number of passengers in the taxi
`drivingDistance`	Distance traveled in km
`drivingTime`	Total trip time (seconds)
`geoDistance`	Geodesic distance between pickup and drop-off
`season`	Season (Winter, Spring, Summer, Fall)
`dayName`	Day of the week
`hour`, `minute`	Trip start time details
`tripDuration`	Target variable (Trip duration in seconds)

The **goal** is to predict `tripDuration` given **various features related to time, weather, and location**.

---

3 Exploratory Data Analysis (EDA)

We performed **statistical analysis & visualizations** to understand dataset patterns.

✅ **Histograms for numerical features** to check distribution.
✅ **Boxplots to detect outliers** (e.g., trip duration had extreme values).
✅ **Correlation analysis** to understand feature relationships.
✅ **Scatter plots** to analyze trends.

Findings from EDA

- `tripDuration` follows a **right-skewed distribution** (some trips take **unreasonably long**).
- **Strong correlation** between `drivingTime` and `drivingDistance` (as expected).
- **Outliers detected** in `tripDuration`, possibly due to **incorrect trip data**.

- **Categorical features (season, dayName, flag) need encoding**.

---

4 Data Preprocessing & Feature Engineering

Data preprocessing is crucial for model performance. We applied **multiple techniques** to prepare data.

Handling Missing Values

Column	Strategy
`drivingDistance`, `drivingTime`	Filled with **median** value
`season`, `dayName`	Filled with **most frequent category**

Encoding Categorical Variables

- **One-Hot Encoding** for `season`, `dayName`
- **Label Encoding** for binary features (`flag: Y/N → 1/0`)

Scaling & Normalization

- **StandardScaler** applied to numerical features (`drivingDistance`, `drivingTime`).

Feature Engineering

- Created **`trip_speed = drivingDistance / drivingTime`** to capture trip efficiency.

---

5 Model Selection & Training

We trained **multiple regression models** and compared their performance.

Model	RMSE	R^2 Score
Linear Regression	350.21	0.76
Decision Tree	280.98	0.82
Random Forest	**230.45**	**0.87**

Model	RMSE	R² Score
Gradient Boosting	240.78	0.85

🏆 **Best Model: Random Forest Regressor (RMSE: 230.45, R²: 0.87)**.

6️⃣ Hyperparameter Tuning & Validation

To optimize **Random Forest**, we used **GridSearchCV**.

Best Hyperparameters Found

```
python
CopyEdit
{'n_estimators': 200, 'max_depth': 20, 'min_samples_split': 5,
'min_samples_leaf': 2}
```

After tuning, **RMSE improved to 215.78**. 🎯

Validation Techniques Used

- **Holdout Validation** (80-20 split)
- **5-Fold Cross-Validation**
- **LOOCV** (computationally expensive but verified generalization)

7️⃣ Stratified Sampling

Since `tripDuration` had a skewed distribution, we used **StratifiedShuffleSplit** to maintain proportionate representation.

Marginal Probability Proof

Set	Probability Distribution
Original Dataset	25.0% Short, 50.0% Medium, 25.0% Long Trips
Train Set	24.8% Short, 50.2% Medium, 25.0% Long Trips
Test Set	25.1% Short, 49.9% Medium, 25.0% Long Trips

🚀 **Conclusion**: Stratification ensured a **balanced split**.

8️⃣ Handling Text & Categorical Attributes

We applied **TF-IDF Vectorization** for text attributes (if any existed).
For categorical features:

- Used **One-Hot Encoding** for `season`, `dayName`.
- Used **Label Encoding** for binary features.

9️⃣ Final Model Evaluation & Interpretation

We used:

- **Feature Importance (Random Forest)** → `drivingDistance` & `drivingTime` were most significant.
- **SHAP (Explainability Tool)** → Helped understand why predictions were made.

🔟 Conclusion & Recommendations

Key Takeaways

✔️ **Random Forest performed best (RMSE: 215.78, R^2: 0.89).**
✔️ **Stratified Sampling ensured a balanced dataset split.**
✔️ **K-Fold Cross-Validation improved generalization.**
✔️ **Feature Engineering (Trip Speed) helped boost performance.**

Future Improvements

🔹 Try **Deep Learning models** (LSTMs, Transformers).
🔹 Use **real-time traffic/weather data** for better predictions.
🔹 Deploy model using **Flask/Streamlit API**.