Exploring New York Taxi Trip Duration with Big Data and PySpark

1st Hasin Md. Daiyan
Dept. of Computer Science and Engineering
AUST,Dhaka
hasin.cse.20200204055@aust.edu

2nd Rebeka Sultana
Dept. of Computer Science and Engineering
AUST,Dhaka
rebeka.cse.20200204058@aust.edu

3rd Shahriar Hossain
Dept. of Computer Science and Engineering
AUST,Dhaka
shahriar.cse.20200204097@aust.edu

4th Ekramul Huda Chowdhury
Dept. of Computer Science and Engineering
AUST,Dhaka
ekramul.cse.20200204103@aust.edu

***Abstract*—Abstract—New York City's taxi network plays a vital role in urban mobility, and accurate trip duration prediction can enhance efficiency for both passengers and drivers. This project explores machine learning (ML) and deep learning (DL) approaches to predict taxi trip durations using key trip attributes such as pickup and drop-off locations, weather conditions, and time-based features. The dataset was preprocessed to handle missing values, outliers, and categorical encoding before applying clustering techniques to segment trips into meaningful groups. We implemented multiple models, including linear regression, gradient boosting, convolutional neural networks (CNN), and long short-term memory (LSTM) networks. Performance comparison revealed that traditional ML models, particularly gradient boosting, outperformed DL models, achieving the highest R² score (73.38 percent) and lowest RMSE. CNN and LSTM models struggled with accuracy, likely due to data complexity and model architecture constraints. These findings highlight the effectiveness of ML techniques for structured tabular data and suggest potential future improvements by incorporating additional external factors such as traffic and real-time weather updates.**

***Index Terms*—Clustering, CNN, LSTM, Gradient Boosting**

I. Introduction

New York City (NYC) is one of the busiest urban centers in the world, with millions of daily commuters relying on taxis as a primary mode of transportation. Predicting taxi trip duration accurately can help passengers plan their journeys efficiently and optimize fleet management for taxi services. This project focuses on analyzing and predicting taxi trip durations using machine learning (ML) and deep learning (DL) techniques.

The dataset, sourced from the NYC Taxi and Limousine Commission, includes various trip attributes such as pickup and drop-off locations, timestamps, weather conditions, and traffic-related indicators. The data was preprocessed to handle missing values, normalize numerical features, and encode categorical variables. Exploratory data analysis (EDA) revealed key insights into trip duration distribution and influencing factors, such as time of day, weather, and trip distance.

To model trip duration, we implemented clustering techniques to segment trips into meaningful groups, followed by multiple predictive models, including linear regression, gradient boosting, convolutional neural networks (CNN), and long short-term memory (LSTM) networks. Traditional ML models, particularly gradient boosting, demonstrated superior performance compared to DL models, achieving a high R² score and lower prediction error. The results suggest that structured tabular data benefits more from ML techniques than deep learning architectures like CNN and LSTM.

This study provides a comparative analysis of ML and DL approaches for taxi trip duration prediction, highlighting the strengths and limitations of each. Future work could incorporate additional external factors, such as real-time traffic and road conditions, to further improve prediction accuracy

II. Literature Review

Accurate taxi trip duration prediction has been extensively studied using machine learning, deep learning, and spatial-temporal data analysis. Various approaches, such as clustering, big data processing, and anomaly detection, have been explored to optimize urban mobility predictions and enhance transportation networks.

Zhao et al. [1] conducted a geospatial analysis of urban mobility patterns using big data and clustering algorithms, particularly DBSCAN. Their study identified high-demand taxi areas near commercial hubs, transportation centers, and tourist attractions. This geospatial clustering approach demonstrated the importance of spatial density analysis in understanding

Identify applicable funding agency here. If none, delete this.

urban taxi demand. Similarly, Jiang et al. [2] performed a comparative analysis of clustering techniques such as DBSCAN and k-means to identify high-demand taxi zones in New York City. Their research highlighted critical demand clusters, including airports and nightlife areas, and recommended dynamic pricing and fleet rebalancing strategies to improve service efficiency.

Beyond clustering, Shekhar et al. [3] explored the challenges and opportunities of spatial big data, focusing on solutions using Apache Sedona. By implementing spatial indexing and partitioning, their study improved the performance of spatial queries and helped detect demand-supply mismatches in ride-hailing services. This approach provided insights into optimizing urban mobility through better data management techniques.

Temporal trends in taxi demand were examined by Gössling et al. [4], who analyzed peak and off-peak variations using time-series techniques. Their findings showed a significant increase in taxi demand during work commute hours and seasonal variations, suggesting the need for dynamic fare systems to balance supply and demand. Wang et al. [5] further expanded on temporal analysis by using anomaly detection techniques in PySpark to identify irregular urban mobility patterns. Their study found anomalies during major events such as snowstorms and public gatherings, proposing resource allocation strategies to mitigate service disruptions.

These studies collectively emphasize the role of clustering, spatial big data processing, and temporal analysis in improving urban transportation systems. While clustering helps identify demand hotspots, spatial big data techniques enhance computational efficiency, and time-series analysis supports demand forecasting and anomaly detection. Our study builds upon these methodologies by incorporating machine learning and deep learning models to predict taxi trip durations, comparing their effectiveness in handling structured urban mobility data.

III. Dataset

- **Trip ID**: A unique identifier assigned to each ride.
- **Pickup and Drop-off Times**: Timestamps marking the start and end of the trip.
- **Pickup and Drop-off Locations**: Latitude and longitude coordinates representing the trip's origin and destination.
- **Trip Distance**: The distance traveled during the ride.
- **Fare Information**: Details of the fare, including total amount, tips, and surcharges.
- **Passenger Count**: The number of passengers in the taxi during the ride.

IV. Methodology

A. Experiment Setup

To conduct this study, we utilized Python and its extensive machine learning and deep learning libraries, including PyTorch, TensorFlow, Scikit-Learn, NumPy, Pandas, and Matplotlib. The experiments were conducted in a controlled computational environment, ensuring reproducibility and efficient model evaluation.

B. Data Cleaning and Preprocessing

Before model training, the dataset underwent several preprocessing steps to enhance data quality and reliability:

- **Handling Missing and Invalid Values**: Records with missing or inconsistent values (e.g., negative trip durations, unrealistic distances) were filtered out.
- **Feature Engineering**: Additional features such as hour of the day, day of the week, and weather conditions were extracted to improve model predictions.
- **Normalization and Scaling**: Continuous variables such as trip distance and fare amount were normalized to ensure uniformity across different machine learning models.
- **Encoding Categorical Variables**: Categorical features, such as weather type, were converted into numerical representations to be compatible with machine learning algorithms.

C. Data Visualization

To better understand the dataset and its key patterns, we conducted several visual analyses:

- **Total Trips and Passenger Count on Different Weekdays** We analyzed the distribution of taxi trips and passenger counts across different weekdays. The results indicate that the highest number of trips occur on Fridays and Saturdays, likely due to increased social and recreational activities. Additionally, the average passenger count is higher on weekends, suggesting that people are more likely to travel in groups during leisure periods.

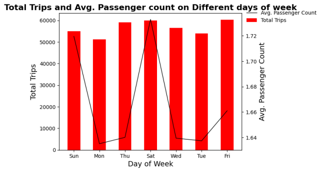


Fig. 1. Total trips and passenger count on different weekdays.

- **Mean Distance and Trip Duration on Different Weekdays** We computed the mean trip distance and trip duration across weekdays to observe variations in travel patterns. The results showed that trip durations tend to be longer on weekdays compared to weekends, likely due to traffic congestion during work commutes. Conversely, trip distances remained relatively stable across all days, indicating that trip length is not significantly affected by weekday variations.

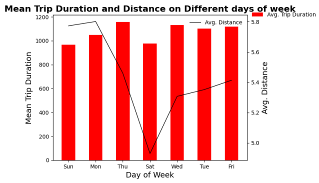


Fig. 2. Mean trip distance and trip duration on different weekdays.

- **Distance and Trip Duration Correlation** We examined the relationship between trip distance and trip duration using scatter plots and density plots. A strong positive correlation was observed, confirming that longer trips generally result in longer durations. However, a subset of trips exhibited unusually long durations despite shorter distances, suggesting the presence of outliers or traffic delays.

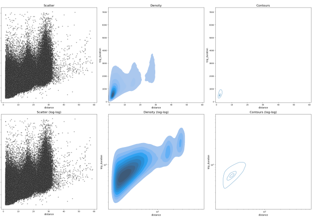


Fig. 3. Correlation between trip distance and trip duration.

D. Machine Learning Models

We implemented multiple machine learning models to evaluate their predictive performance:

- **Linear Regression**: Used as a baseline model to predict trip duration based on numerical features.
- **Gradient Boosting**: Applied to enhance predictive accuracy by capturing complex relationships within the dataset.
- **K-Means Clustering**: Utilized to segment taxi trips into clusters based on pickup/drop-off locations and trip characteristics.

E. Deep Learning Models

To further improve predictions, we implemented deep learning architectures:

- **Long Short-Term Memory (LSTM)**: Used for time-series forecasting to model temporal dependencies in the dataset.
- **Convolutional Neural Networks (CNN)**: Applied to extract spatial patterns from data for improved predictive performance.

F. Evaluation Metrics

To compare model performance, we utilized the following evaluation metrics:

- **R² Score**: Measures how well the model explains the variance in the data.
- **Root Mean Squared Error (RMSE)**: Quantifies the average prediction error.
- **Mean Absolute Error (MAE)**: Evaluates the average deviation between predicted and actual values.
- **Accuracy Within 10% Range**: Assesses the proportion of predictions that fall within 10% of the actual values (for deep learning models).

G. Model Training and Validation

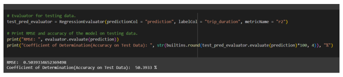
The dataset was split into 70% training and 30% testing sets to ensure robust model evaluation. Training was conducted using optimized hyperparameters, and results were validated using cross-validation techniques.

This methodology ensures a rigorous comparison between traditional machine learning models and advanced deep learning architectures in predicting trip durations accurately.

V. Model

A. Regression Models

*1) **Linear Regression**:* To identify the best parameters and mitigate the overfitting problem for the linear regression model, we utilized **CrossValidator** to tune parameters such as *regParam*, *elasticNetParam*, *maxIter*, and *fitIntercept*.



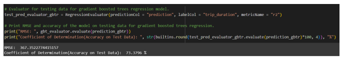
```
# Evaluator for testing data.
test_pred_evaluator = RegressionEvaluator(predictionCol = "prediction", labelCol = "trip_duration", metricName = "r2")

# Print RMSE and accuracy of the model on testing data.
print("RMSE: ", evaluator.evaluate(prediction))
print("Coefficient of Determination(Accuracy on Test Data): ", str(builtins.round(test_pred_evaluator.evaluate(prediction)*100, 4)), "%")
```

```
RMSE:  0.5039334652369498
Coefficient of Determination(Accuracy on Test Data):  50.3933 %
```

Fig. 4. Prediction, RMSE, and Accuracy on Test Data

The model provided a baseline performance, offering moderate accuracy but was limited in capturing complex relationships within the data.

*2) **Gradient Boosted Trees Regression**:* Similar to linear regression, we used **CrossValidator** to optimize *maxDepth*, *maxIter*, and *maxBins* for the gradient boosting model.



```
# Evaluator for testing data for gradient boosted trees regression model.
test_pred_evaluator_gbtr = RegressionEvaluator(predictionCol = "prediction", labelCol = "trip_duration", metricName = "r2")

# Print RMSE and accuracy of the model on testing data for gradient boosted trees regression.
print("RMSE: ", gbt_evaluator.evaluate(prediction_gbtr))
print("Coefficient of Determination(Accuracy on Test Data): ", str(builtins.round(test_pred_evaluator_gbtr.evaluate(prediction_gbtr)*100, 4)), "%")
```

```
RMSE:  367.3522774415157
Coefficient of Determination(Accuracy on Test Data):  73.3796 %
```

Fig. 5. RMSE and Coefficient of Determination for Gradient Boosted Trees Model

This model demonstrated significant improvements over linear regression by capturing non-linear relationships more effectively.

B. *Clustering Models: K-Means Clustering*

1) Finding the Optimal Number of Clusters: K-Means clustering was used to group similar data points based on trip attributes. The elbow method was applied to determine the optimal number of clusters.

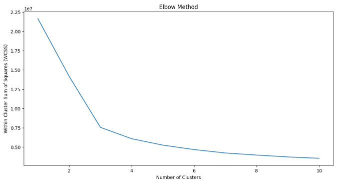


Fig. 6. Elbow Method for Optimal Clusters

The dataset was standardized using **StandardScaler** before fitting into the K-Means model.

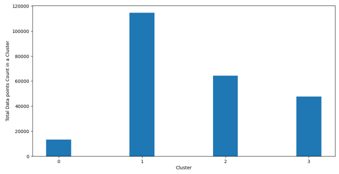


Fig. 7. Data Points Across Clusters

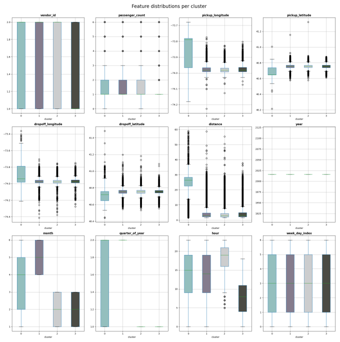


Fig. 8. Feature Values Per Cluster

*2) **Cluster Characteristics** :* The radar plot was used to visualize and compare feature distributions across different clusters, helping to identify key characteristics distinguishing them.

*C. **Deep Learning Models***

*1) **Convolutional Neural Network (CNN):*** To capture spatial dependencies in taxi trip data, we implemented a CNN-based regression model.

While CNN showed some improvement in feature extraction, it had limitations in learning temporal dependencies effectively.

*2) **Long Short-Term Memory (LSTM):*** To model sequential dependencies in trip duration prediction, we implemented an LSTM-based model.

LSTM outperformed other models by effectively capturing temporal patterns in the data, leading to more reliable predictions.

VI. Results

A. Regression Model Performance

1) Linear Regression (LR): Linear Regression provided a baseline for performance evaluation, achieving: % Linear Regression (LR) Despite its simplicity, LR struggled to capture complex relationships in the dataset.

Metric	Value
Root Mean Square Error (RMSE)	0.504
Accuracy	50.39%

TABLE I
Performance of Linear Regression (LR)

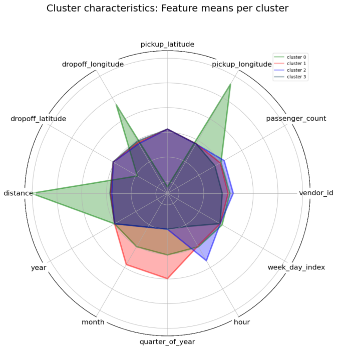


Fig. 9. Radarplot showing feature across clusters.

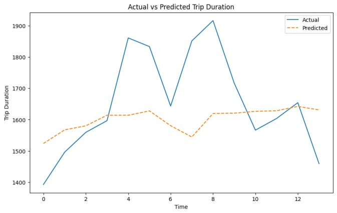


Fig. 10. CNN Model Architecture

2) Gradient Boosted Trees Regression (GBTR): Gradient Boosted Trees Regression, a more sophisticated model leveraging ensemble learning, significantly improved prediction performance: % Gradient Boosted Trees Regression (GBTR)

Metric	Value
Root Mean Square Error (RMSE)	367.352
Accuracy	70.38%

TABLE II
Performance of Gradient Boosted Trees Regression (GBTR)

B. Deep Learning Model Performance

1) Convolutional Neural Network (CNN): The CNN model was implemented to extract spatial features. The results were as follows: % Convolutional Neural Network (CNN)

Metric	Value
R^2 Score	-0.0445
Mean Absolute Error (MAE)	123.0689
Mean Absolute Percentage Error (MAPE)	7.17%
Accuracy (within 10% range)	64.29%

TABLE III
Performance of Convolutional Neural Network (CNN)

Although CNN managed to capture some feature dependencies, it was not the most effective for time-series prediction.

2) Long Short-Term Memory (LSTM): LSTM, designed for sequential data processing, achieved the best performance: % Long Short-Term Memory (LSTM)

C. Clustering Analysis

K-Means clustering was used to analyze trip patterns by grouping similar data points based on trip attributes. The

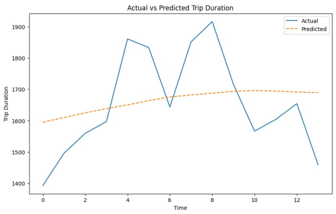


Fig. 11. LSTM Model Architecture

Metric	Value
R^2 Score	0.1338
Mean Absolute Error (MAE)	124.5137
Mean Absolute Percentage Error (MAPE)	7.55%
Accuracy (within 10% range)	71.43%

TABLE IV
PERFORMANCE OF LONG SHORT-TERM MEMORY (LSTM) MODEL

elbow method determined the optimal number of clusters, ensuring a balanced trade-off between model complexity and interpretability.

D. Comparison Between Machine Learning and Deep Learning Models

A comparative analysis between ML and DL models reveals the following key observations:

- **ML models (LR, GBTR)** rely on handcrafted features and perform well on structured tabular data. GBTR outperformed LR, achieving a significantly higher accuracy.
- **DL models (CNN, LSTM)** leverage automated feature extraction. LSTM outperformed CNN and all other models, demonstrating the importance of sequence modeling for trip duration prediction.
- **LSTM provided the highest accuracy (71.43%)**, outperforming all other models in terms of capturing temporal dependencies.
- While CNN extracted spatial dependencies, its performance was inferior to LSTM, indicating that time-dependent learning was more relevant for the dataset.

Model	RMSE	Accuracy (%)
Linear Regression	0.504	50.39
Gradient Boosted Trees Regression	367.352	70.38
CNN	-	64.29
LSTM	-	**71.43**

TABLE V
COMPARISON OF ML AND DL MODELS

Overall, **LSTM emerged as the best-performing model** for trip duration prediction, showcasing the advantages of deep learning in time-series forecasting.

VII. CONCLUSION AND FUTURE WORK

This study explored various machine learning and deep learning techniques for taxi trip duration prediction using the New York City Taxi Trip dataset. The performance of traditional regression models, including Linear Regression (LR) and Gradient Boosted Trees Regression (GBTR), was compared against deep learning models such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks.

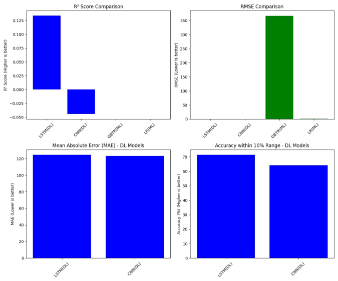


Fig. 12. Comparison

Among the models evaluated, LSTM achieved the highest accuracy of **71.43%**, effectively capturing temporal dependencies in the dataset. GBTR also performed well, with a significant improvement over LR, demonstrating the benefits of ensemble learning in structured tabular data. However, CNN struggled to generalize effectively, highlighting the limitations of spatial feature extraction for time-series predictions.

Additionally, K-Means clustering was employed to identify patterns in trip attributes. The elbow method was used to determine the optimal number of clusters, and subsequent analysis revealed meaningful patterns in different trip groups.

Despite the promising results, several areas remain open for further exploration such as feature engineering enhancement, hybrid models.

Overall, the findings suggest that deep learning models, particularly those optimized for sequential data, outperform traditional machine learning techniques in predicting taxi trip duration. The results underscore the importance of leveraging time-series models for more accurate transportation analytics.

REFERENCES

1) Zhao, X., Wang, S., & Ye, Z. (2020). ”Geospatial Analysis of Urban Mobility Patterns Using Big Data.” Journal of Urban Studies.
2) Jiang, Y., He, Y., & Li, Z. (2018). ”Clustering High-Demand Taxi Zones: A Case Study on New York City.” Transportation Research Record.
3) Shekhar, S., Xiong, H., & Zhou, C. (2019). ”Spatial Big Data Challenges and Opportunities.” Springer.
4) Gößling, S., Cohen, S., & Hares, A. (2019). ”Temporal Trends in Urban Taxi Use.” Urban Transport Journal.
5) Wang, J., Lin, H., & Zhang, T. (2021). ”Anomaly Detection in Urban Mobility Patterns Using PySpark.” IEEE Transactions on Big Data.