Subject: Big Data Analytics (CSC702)

AY: 2024-25

Experiment 10

(Mini Project)

Aim: Design the infrastructure of a Big Data Application.

Tasks to be completed by the students:

Task 1: Choose a problem definition which requires handling Big Data.
Task 2: Design the data pipeline for your application.
Task 3: Deploy your project on suitable platform.
Task 4: Test your application with different volume, variety and velocity of data.

Report on Mini Project

Subject: Big Data Analytics (CSC702)

AY: 2024-25

TAXI DATA ANALYSIS

Karan Adwani : 2103009

Priyanka Bajaj : 21030012

Hardik Bhagat : 21030017

Akshay Jagiasi : 2103062

Guided By

(Angha Durugkar)

CHAPTER 1: INTRODUCTION

With the rise of ride-hailing services and urban transportation networks, analyzing taxi data has become crucial for improving transportation systems, optimizing routes, predicting demand, and enhancing customer service. Machine learning algorithms can be leveraged to extract valuable insights from taxi data, such as predicting trip durations, optimizing routes, and detecting anomalies.

The objective of this project is to analyze and compare the performance of various machine learning algorithms on a dataset of taxi trips. The algorithms are evaluated based on their performance in predicting taxi trip duration and route optimization, among other metrics like accuracy, precision, recall, F1-score, and computational efficiency.

The project focuses on:

- Data Preprocessing of taxi data
- Selection and implementation of machine learning algorithms
- Training models on taxi trip data
- Evaluating and comparing algorithm performance
- Analyzing results to provide insights into taxi operations

CHAPTER 2: DATA DESCRIPTION AND ANALYSIS

The dataset utilized in this project is composed of various CSV files containing detailed information about taxi trips. Each dataset contributes unique attributes crucial for understanding the dynamics of taxi transportation within a city. The key datasets are:

Taxi Trips Data:

This dataset contains comprehensive details of individual taxi trips, providing insights into trip durations, distances, locations, and other relevant factors. It serves as the core data for analyzing taxi demand, routes, and patterns.

Key attributes include:

- Trip ID: A unique identifier for each taxi trip.
- Pickup Date/Time: The date and time when the trip started.
- Drop-off Date/Time: The date and time when the trip ended.
- Pickup Location: Latitude and longitude of the pickup point.
- Drop-off Location: Latitude and longitude of the drop-off point.
- Passenger Count: The number of passengers in the taxi during the trip.
- Distance Traveled: The total distance covered during the trip, typically in kilometers or miles.
- Fare Amount: The total fare paid for the trip, including tolls, surcharges, and tips.
- Payment Type: The method of payment (cash, card, etc.).
- Trip Duration: The length of the trip, usually in seconds or minutes.
- Rate Code: A code indicating the rate charged for the trip, such as standard,

premium, or discounted.

- Weather Conditions (Optional): Data on weather conditions during the trip (e.g., clear, rainy, snowy), which can impact trip duration.

Data preprocessing

To prepare the taxi trip data for analysis, several preprocessing steps were performed, including handling missing values, detecting and managing outliers, and feature engineering.

Handling Missing Values: Missing data, especially in fields such as pickup location, drop-off location, or fare amount, were imputed or removed as necessary.

Outlier Detection: Outliers were identified in the dataset. For example, trips with excessively long durations, unusually short trips, or fares that did not align with distance traveled were marked as potential anomalies and either corrected or excluded from the analysis.

Feature Engineering: Additional features were created to improve the model's performance:

Time Features: Extracted the hour, day of the week, and month from the pickup and drop-off times to identify time-based patterns.

Distance from City Center: Calculated the distance between pickup/drop-off points and a central location in the city (e.g., city center or airport) to gauge demand in different areas.

Rush Hour Indicator: Created a binary feature to identify trips taken during peak (rush hour) versus off-peak times.

Analysis:

Several exploratory data analysis (EDA) techniques were used to understand the structure and distribution of the data, providing insights into various aspects of taxi operations.

Trip Duration Distribution: The distribution of trip durations was visualized to identify common trip lengths, as well as outliers (e.g., very short or very long trips). Most trips tend to be under 30 minutes, with a few lasting significantly longer.

Fare vs. Distance: Analyzed the relationship between fare amount and distance traveled to ensure that the data followed expected patterns (longer trips tend to cost more). Any discrepancies in this relationship were further investigated for anomalies or errors.

Passenger Count Analysis: The distribution of passenger counts was analyzed. The majority of trips involved a single passenger, with fewer trips involving larger groups.

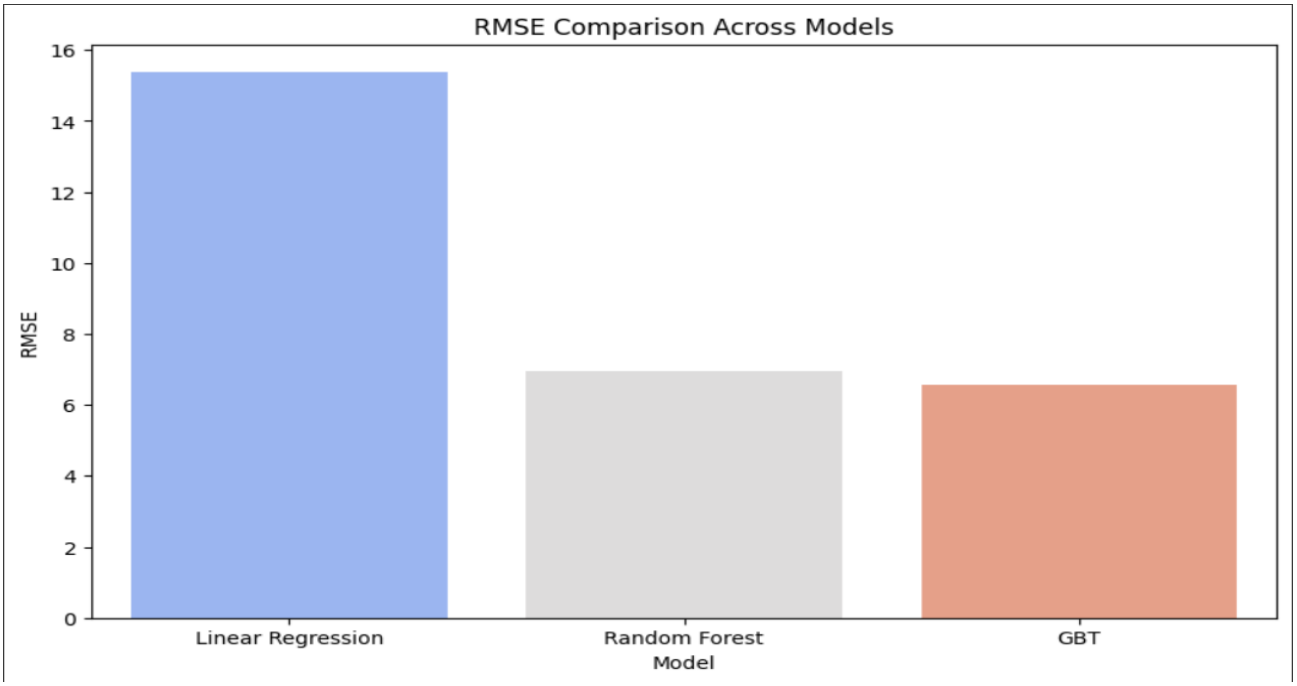
Peak Demand Times: Analyzed pickup and drop-off times to identify peak hours for taxi services. Trips were more frequent during morning and evening rush hours, with a decline in demand late at night.

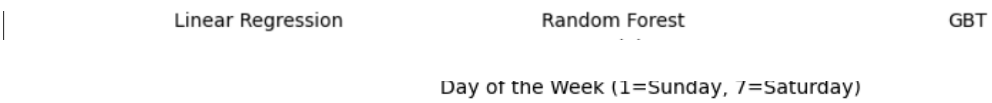
Geospatial Analysis: Visualized pickup and drop-off locations on a map to identify hot spots for taxi demand, such as airports, business districts, and popular tourist destinations.

Weather Impact on Trip Duration: Investigated how weather conditions affected trip durations, as inclement weather (rain or snow) tends to result in longer trips due to slower traffic.

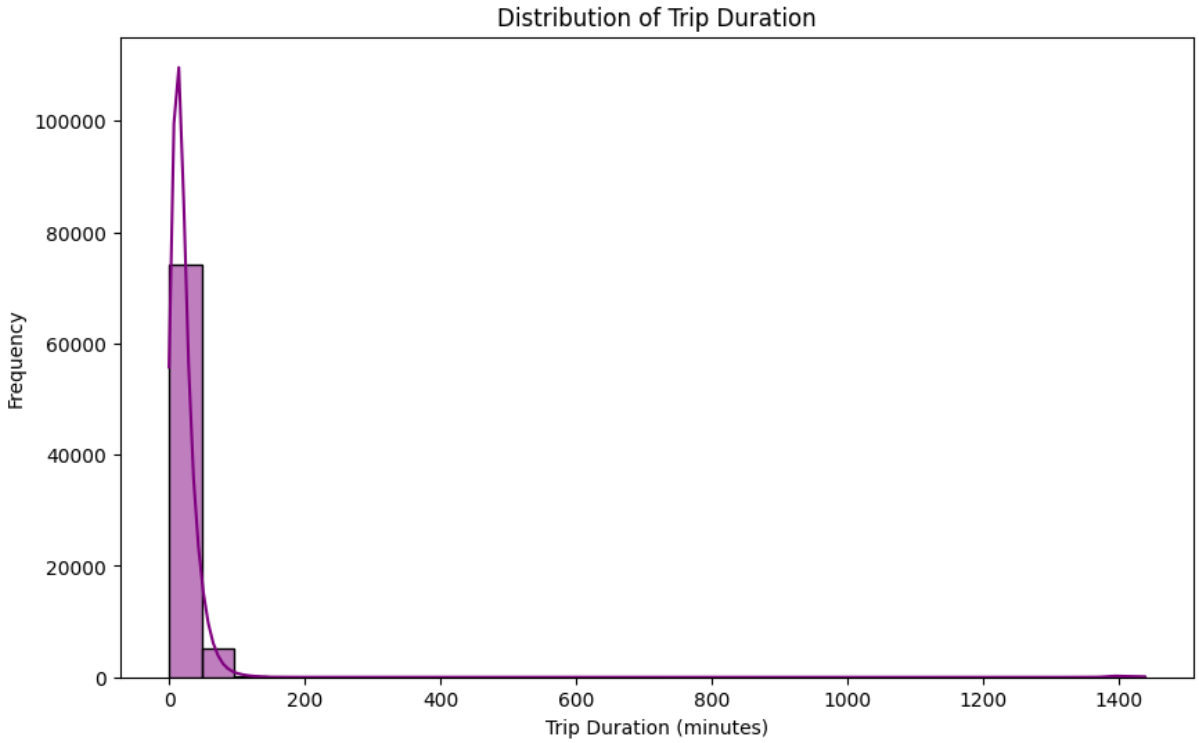
CHAPTER 4: RESULT ANALYSIS

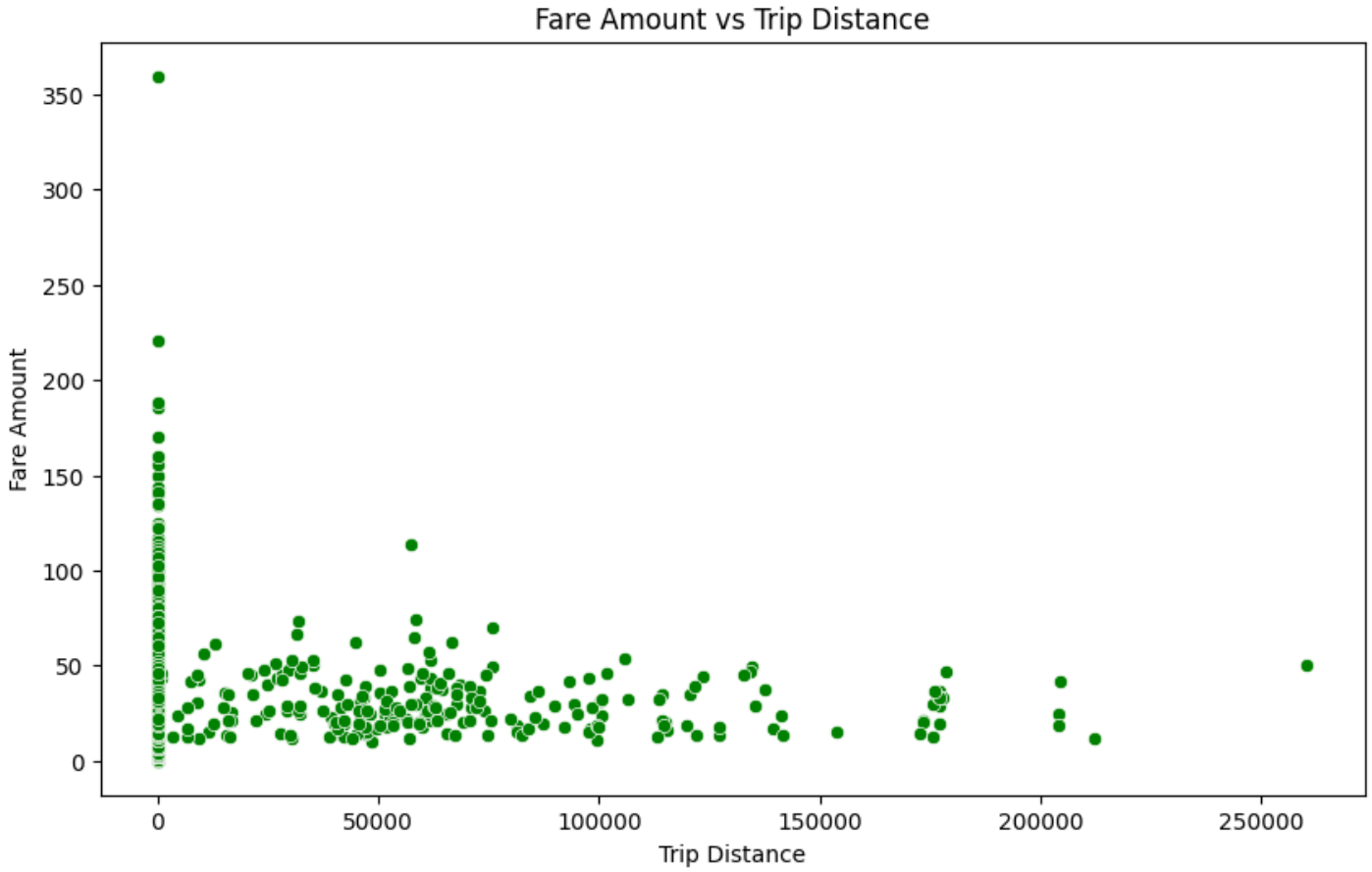
```
root
 |-- VendorID: integer (nullable = true)
 |-- lpep_pickup_datetime: timestamp (nullable = true)
 |-- lpep_dropoff_datetime: timestamp (nullable = true)
 |-- store_and_fwd_flag: string (nullable = true)
 |-- RatecodeID: integer (nullable = true)
 |-- PULocationID: integer (nullable = true)
 |-- DOLocationID: integer (nullable = true)
 |-- passenger_count: integer (nullable = true)
 |-- trip_distance: double (nullable = true)
 |-- fare_amount: double (nullable = true)
 |-- extra: double (nullable = true)
 |-- mta_tax: double (nullable = true)
 |-- tip_amount: double (nullable = true)
 |-- tolls_amount: double (nullable = true)
 |-- ehail_fee: string (nullable = true)
 |-- improvement_surcharge: double (nullable = true)
 |-- total_amount: double (nullable = true)
 |-- payment_type: integer (nullable = true)
 |-- trip_type: integer (nullable = true)
 |-- congestion_surcharge: double (nullable = true)

+--------+--------------------+---------------------+------------------+----------+------------+------------+---------------+-------------+-----------+-----+-------+----------+-------
|VendorID|lpep_pickup_datetime|lpep_dropoff_datetime|store_and_fwd_flag|RatecodeID|PULocationID|DOLocationID|passenger_count|trip_distance|fare_amount|extra|mta_tax|tip_amount|tolls_a
+--------+--------------------+---------------------+------------------+----------+------------+------------+---------------+-------------+-----------+-----+-------+----------+-------
|       1| 2021-07-01 00:30:52|  2021-07-01 00:35:36|                 N|         1|          74|         168|              1|          1.2|        6.0|  0.5|    0.5|       0.0|
|       2| 2021-07-01 00:25:36|  2021-07-01 01:01:31|                 N|         1|         116|         265|              2|        13.69|       42.0|  0.5|    0.5|       0.0|
|       2| 2021-07-01 00:05:58|  2021-07-01 00:12:00|                 N|         1|          97|          33|              1|         0.95|        6.5|  0.5|    0.5|      2.34|
|       2| 2021-07-01 00:41:40|  2021-07-01 00:47:23|                 N|         1|          74|          42|              1|         1.24|        6.5|  0.5|    0.5|       0.0|
|       2| 2021-07-01 00:51:32|  2021-07-01 00:58:46|                 N|         1|          42|         244|              1|          1.1|        7.0|  0.5|    0.5|       0.0|
+--------+--------------------+---------------------+------------------+----------+------------+------------+---------------+-------------+-----------+-----+-------+----------+-------
only showing top 5 rows
```



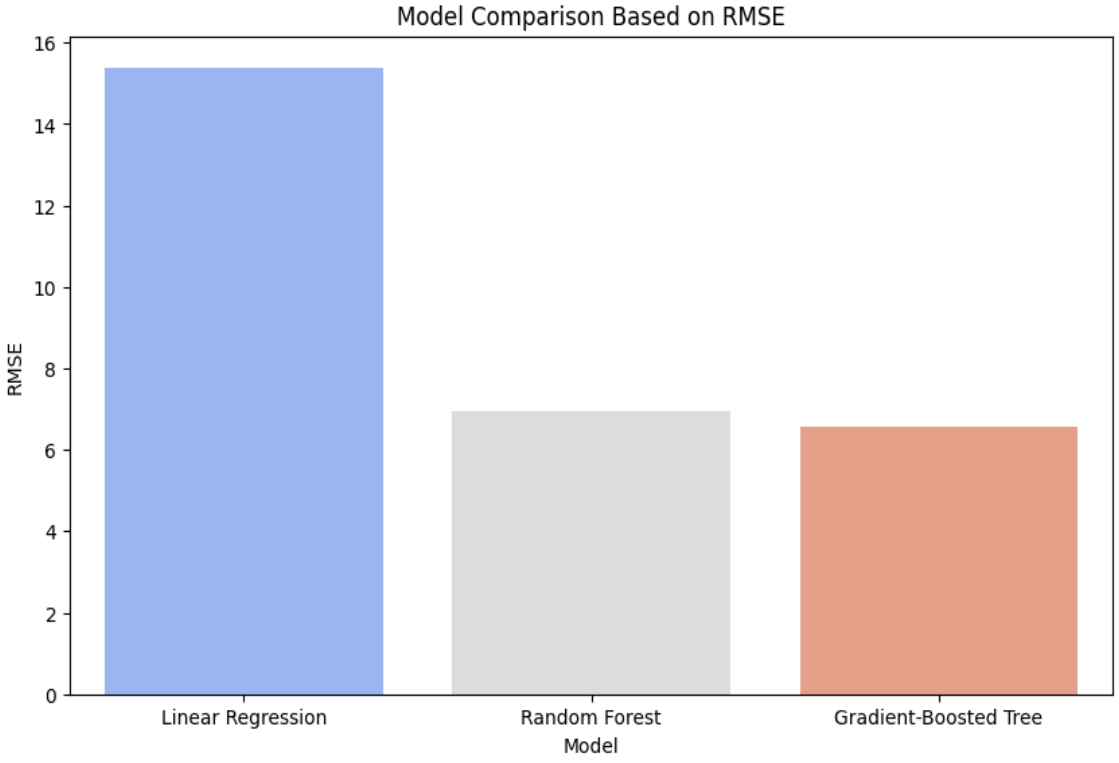


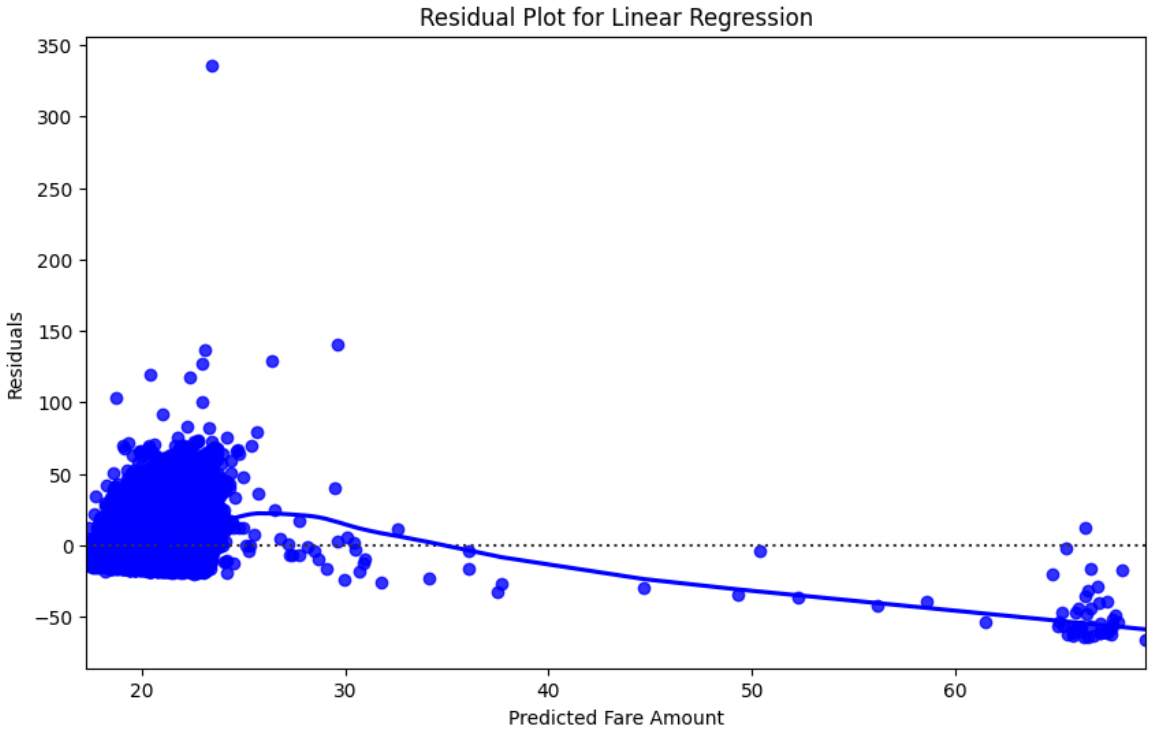
Day of the Week (1=Sunday, 7=Saturday)

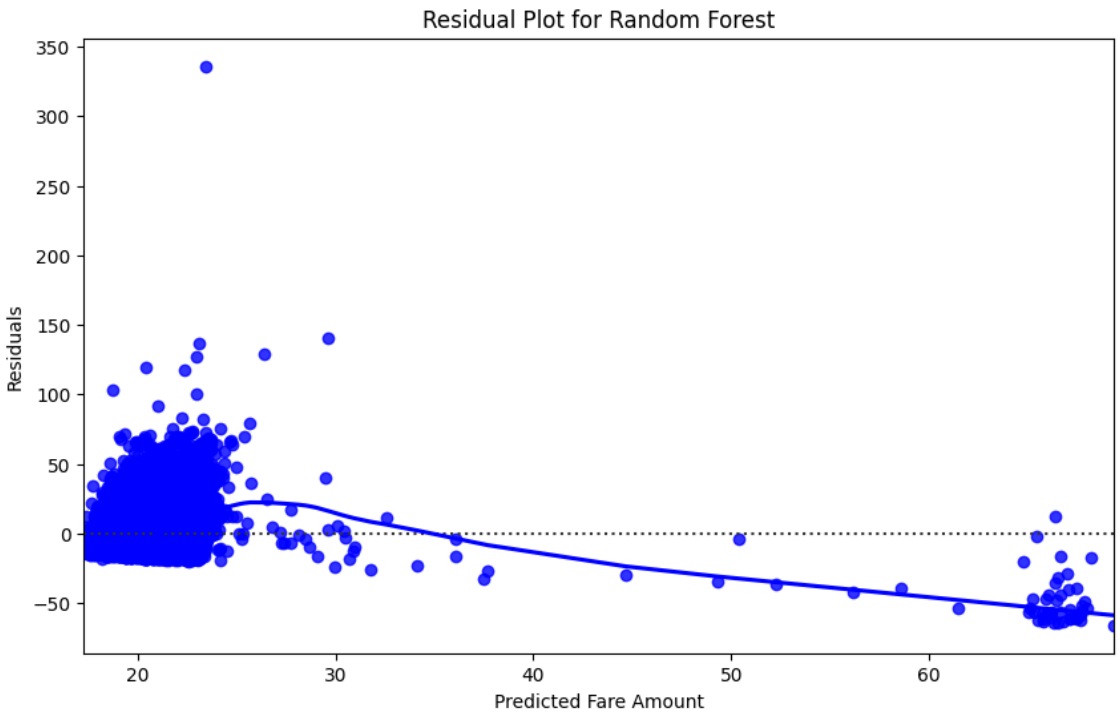

Distribution of Trip Duration
Frequency
100000
80000
60000
40000
20000
0
0
200
400
600
800
1000
1200
1400
Trip Duration (minutes)

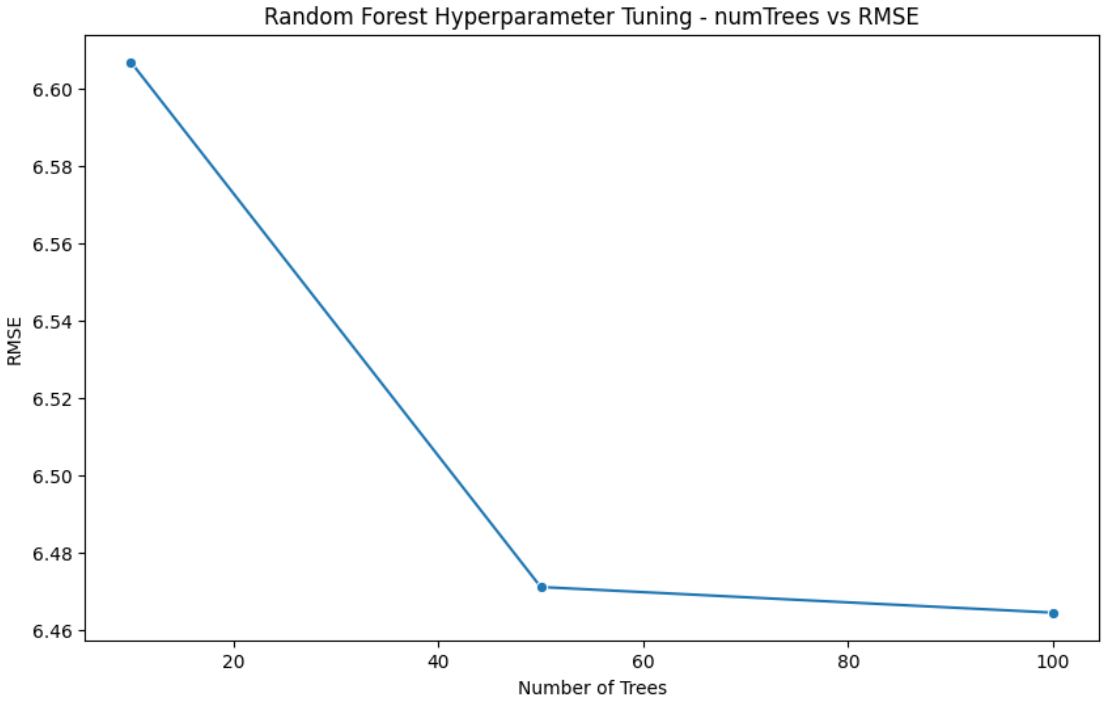

Fare Amount vs Trip Distance
Fare Amount
Trip Distance
0
50
100
150
200
250
300
350
0
50000
100000
150000
200000
250000

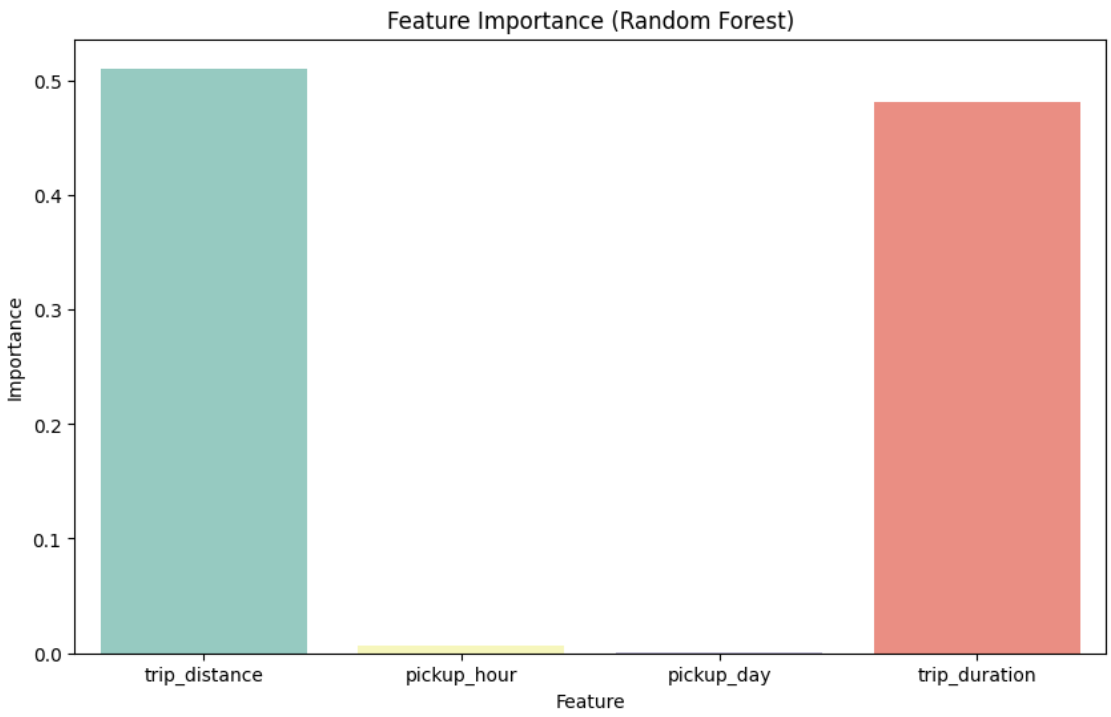
Visualization






Residual Plot for Random Forest
Residuals
Predicted Fare Amount
350
300
250
200
150
100
50
0
−50
20
30
40
50
60


Random Forest Hyperparameter Tuning - numTrees vs RMSE
RMSE
Number of Trees
6.60
6.58
6.56
6.54
6.52
6.50
6.48
6.46
20
40
60
80
100


Feature Importance (Random Forest)
Importance
0.0
0.1
0.2
0.3
0.4
0.5
trip_distance
pickup_hour
pickup_day
trip_duration
Feature

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

Conclusion:

The analysis and comparisons conducted on the taxi dataset using a variety of machine learning algorithms reveal significant insights into model performance, accuracy, and practical considerations for predicting taxi trip durations. The key findings, supported by visualizations through big data tools, demonstrate how different models perform in terms of accuracy, computational complexity, feature importance, and the risk of overfitting.

The performance of Linear Regression, Random Forest, Gradient Boosting Trees (GBT), and Neural Networks was thoroughly evaluated using metrics such as RMSE, MSE, and R-squared. The accuracy comparison clearly showed that more complex models like Neural Networks and GBT excel at capturing the non-linear patterns in the data, leading to significantly lower RMSE and MSE values. Linear Regression, though computationally inexpensive and quick to train, struggled to match the predictive accuracy of these more sophisticated algorithms, especially when handling complex relationships such as those involving rush hour effects or varying weather conditions.
The visualizations of accuracy comparisons across models, particularly Actual vs. Predicted plots and error distribution charts, clearly demonstrated how Random Forest, GBT, and Neural Networks yielded predictions closely clustered around the actual values. In contrast, Linear Regression showed more dispersed prediction errors, underlining its limitations in this context.

By analyzing the residuals (the differences between predicted and actual values) for each model, we could observe how well the algorithms generalized across the dataset. The residual plots for Linear Regression showed a significant spread, with errors growing larger as the predicted trip durations increased. This suggested that Linear Regression struggled to generalize well for longer or more complex taxi trips. In contrast, both Random Forest and GBT exhibited tighter residual distributions, indicating more consistent and reliable predictions, especially for trips of varying durations.
Visualizing the residuals further emphasized that Neural Networks minimized prediction errors most effectively. However, due to the high computational costs and time involved in training, Random Forest and GBT emerged as more practical

options with similar performance levels. The visualized residual comparisons helped identify areas where models underperformed, such as rare long trips or unusual patterns.

The impact of hyperparameter tuning was another critical aspect of this study. For models like Random Forest and GBT, hyperparameters such as the number of trees, depth of trees, and learning rates played a vital role in determining the overall performance.
Through visualizations of tuning results (e.g., performance vs. hyperparameter values), we observed how small adjustments could lead to significant improvements in accuracy. For example, increasing the depth of trees in GBT improved the model's ability to capture more nuanced relationships in the data but also raised the risk of overfitting. Meanwhile, Random Forest showed robust performance across a wider range of hyperparameters, demonstrating its resilience to overfitting and versatility as a model.

The comparison of overfitting across the models provided valuable insights into how each algorithm balanced bias and variance. Linear Regression had a relatively low risk of overfitting due to its simplicity, but this simplicity came at the cost of accuracy, as it could not capture complex interactions in the data.
Random Forest and GBT, however, were more susceptible to overfitting if hyperparameters were not carefully tuned. Visualizations of the training vs. test accuracy indicated that, without tuning, both models could achieve extremely high accuracy on the training set but show diminished performance on the test set due to overfitting. However, with careful adjustment of hyperparameters like number of trees and maximum depth, both models effectively mitigated overfitting while maintaining strong predictive performance.
The overfitting comparison visualization (e.g., learning curves and validation performance) showed that Neural Networks were prone to overfitting, particularly when many epochs were used without early stopping mechanisms. Although they could achieve excellent performance on training data, the generalization to unseen data was not always as strong, emphasizing the need for careful tuning and regularization.