

**UNIVERSITY OF ENERGY AND NATURAL RESOURCES**

**RIDE EXPRESS PREDICTIVE MODELING**

**BY**

**ADINKRAH YEBOAH JUDITH (UEB3228822)**

**MENSAH ANASTASIA AKYAMAA (UEB3221720)**

**OSEI BOATENG ISSABELLA (UEB3207820)**

**CUDJOE BOAFO BENEDICT (UEB3246622)**

**BABA AHMED DEEDAT** **(UEB3215320)**

**A PROJECT WORK SUBMITTED TO THE DEPARTMENT OF INFORMATION TECHNOLOGY AND DECISION SCIENCES**

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE CERTIFICATE OF

**BSC IN INFORMATION TECHNOLOGY**

**SCHOOL OF SCIENCES**

# DECLARATION

We ADINKRAH YEBOAH JUDITH, MENSAH ANASTASIA AKYAMAA, OSEI BOATENG ISSABELLA,CUDJOE BOAFO BENEDICT and BABA AHMED DEEDAT now we declare that this project is our work and, to the best of our knowledge. It contains no material previously published by another person, unless proper acknowledgment is provided within the text.

ADINKRAH YEBOAH JUDITH …………… ……………..

UEB3228822 Signature Date

MENSAH ANASTASIA AKYAMAA …………… ……………..

UEB3221720 Signature Date

OSEI BOATENG ISSABELLA …………… ……………..

UEB3207820 Signature Date

CUDJOE BOAFO …………… ………………

UEB3246622 Signature Date

BABA AHMED DEEDAT …………… ……………..

UEB3215320 Signature Date

Mr. MENSAH JACOB …………… ……………..

Supervisor Signature Date

Dr. PETER APPIAHENE …………… ……………..

Head of Department Signature Date

# ACKNOWLEDGEMENT

God bless you, and thank everyone for the life we have received on this journey. We really appreciate our parents/guardians for the continuous support and sacrifices made while in school. We wish to acknowledge our supervisor Mr. Mensah Jacob for his invaluable guidance, patience and encouragement which were a hallmark of the completion of this work We are really thankful for his wisdom and mentorship.

# DEDICATION

Our work is for our family. Their faithfulness, love, and care for us is what has given us the courage to carry on We also extend to the supportive faculty of Information Technology and Decision Sciences department, whose tutelage has guided our academic journey. Finally, we also want to dedicate this sentiment towards our friends and colleagues in the department whose companionship so enriched that very same journey.

# Abstract

To have a better experience and utilization of resources as well as improve service delivery, this study focuses on predicting the estimated timeliness for each ride of the express vehicle. The duration of the ride express could affect the level of services offered at any particular station. This study embarks on the use of complex machine learning approaches, capable of predicting the duration of each trip made by a ride express. In their complexity handling in real-life situations, it is important not to ignore existing regression-based approaches; this examination of this proposal commences with this point. We then propose a machine-learning framework that supports historical trip data, cleans and preprocesses it, and employs both Linear Regression and Random Forest Regression models to predict trip durations. Our evaluation metrics, such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-squared Score (R²), demonstrate the effectiveness of our models. For Linear Regression, the model achieved the following results: MSE = 256.34, RMSE = 16.01, MAE = 12.53, MAPE = 9.12%, and R² = 0.85. Similarly, for the Random Forest Regression model, the performance was notably better, with MSE = 198.45, RMSE = 14.08, MAE = 10.87, MAPE = 7.56%, and R² = 0.92.

These findings indicate that our machine learning algorithms, particularly Random Forest Regression, greatly surpass traditional techniques, producing more precise and dependable forecasts. This advancement leads to better passenger service operations through improved trip prediction methodology.

Table of Contents

[DECLARATION 2](#_Toc180266839)

[ACKNOWLEDGEMENT 4](#_Toc180266840)

[DEDICATION 5](#_Toc180266841)

[Abstract 6](#_Toc180266842)

[List Abbreviations 10](#_Toc180266843)

[Chapter 1: Introduction 11](#_Toc180266844)

[1.0 Background 11](#_Toc180266845)

[1.1 Scope of the Proposed System 12](#_Toc180266846)

[1.2 Problem Statement 13](#_Toc180266847)

[1.3 Objective 13](#_Toc180266848)

[1.3.0 General Objectives 13](#_Toc180266849)

[1.3.1 Specific Objectives 14](#_Toc180266850)

[1.4 Justification 14](#_Toc180266851)

[1.5 Significance/Importance 14](#_Toc180266852)

[CHAPTER 2: LITERATURE REVIEW 16](#_Toc180266853)

[2.0 Introduction 16](#_Toc180266854)

[2.1.1 Advantages of Computer-Based Systems. 18](#_Toc180266855)

[2.2 Overview of Ride Express 19](#_Toc180266856)

[2.3 Types of Ride Express Data 20](#_Toc180266857)

[2.3.1 Trip Data 20](#_Toc180266858)

[2.3.2 Driver Data 20](#_Toc180266859)

[2.3.3 User Data 20](#_Toc180266860)

[2.3.4 Operational Data 21](#_Toc180266861)

[2.4 Ride Express Data Prediction 21](#_Toc180266862)

[2.5 Advantages of Ride Express Data Prediction 22](#_Toc180266863)

[2.6 Existing Ride Express Prediction 22](#_Toc180266864)

[CHAPTER THREE: METHODOLOGY 23](#_Toc180266865)

[3.0 Introduction 23](#_Toc180266866)

[3.1 Machine Learning Life Cycle 24](#_Toc180266867)

[3.2 Data Collection 25](#_Toc180266868)

[3.3 Data Preprocessing 26](#_Toc180266869)

[3.4 Model Selection 27](#_Toc180266870)

[3.5 Model Training 34](#_Toc180266871)

[3.6 Evaluation 35](#_Toc180266872)

[3.7 Software Development Life Cycle 36](#_Toc180266873)

[CHAPTER FOUR: DESIGN AND IMPLEMENTATION 39](#_Toc180266874)

[4.0 Introduction 39](#_Toc180266875)

[4.0.1 Interface 39](#_Toc180266876)

[4.0.2 Sign up Page 39](#_Toc180266877)

[4.0.3 Login Page 39](#_Toc180266878)

[4.0.4 Home Screen 40](#_Toc180266879)

[4.0.5 Predict Trip Screen 40](#_Toc180266880)

[4.0.6 About Screen 40](#_Toc180266881)

[4.0.7 Rate Rider Page 40](#_Toc180266882)

[4.1 Dashboard 41](#_Toc180266883)

[4.1.1 Filter Section 41](#_Toc180266884)

[4.1.2 Summary Section 41](#_Toc180266885)

[4.1.3 Map Section 42](#_Toc180266886)

[4.1.4 Overall Functionality 42](#_Toc180266887)

[4.2 Testing 43](#_Toc180266888)

[4.3 Levels of Testing 44](#_Toc180266889)

[CHAPTER FIVE: SUMMARY, RECOMMENDATIONS AND CONCLUSION 45](#_Toc180266890)

[5.0 Introduction 45](#_Toc180266891)

[5.1 Summary 45](#_Toc180266892)

[5.2 Recommendations 46](#_Toc180266893)

[References 49](#_Toc180266894)

[APPENDICES 53](#_Toc180266895)

List of Figures

[Figure 1 Machine Learning Process 24](#_Toc180266686)

[Figure 2 Mean Absolute Error 29](#_Toc180266687)

[Figure 3 Mean Absolute Percentage Error 29](#_Toc180266688)

[Figure 4 Mean Squared Error 30](#_Toc180266689)

[Figure 5 R2 Score 30](#_Toc180266690)

[Figure 6 Root Mean Squared Error 30](#_Toc180266691)

[Figure 7 Decision Tree 31](#_Toc180266692)

[Figure 8 Gradient Boosting Regressor 31](#_Toc180266693)

[Figure 9 Linear Regression 32](#_Toc180266694)

[Figure 10 Random Forest Regressor 32](#_Toc180266695)

[Figure 11 Dashboard 41](#_Toc180266696)

# List Abbreviations

1. Mean Squared Error – MSE

2. Root Mean Squared Error – RMSE

3. R Squared Score –R2

4. Mean Absolute Error – MAE

5. Mean Absolute Percentage Error – MAPE

6. Software Development Life Cycle - SDLC

# Chapter 1: Introduction

## 1.0 Background

Big Data applications are changing how companies collect, process and analyze data. This makes the functions of Big Data a key asset in this digital era with its application being done to people-generated or automated data collected by companies for commercial purpose. This transformation, which started in the early 2000s, allowed major tech companies to leverage Big Data for important data-driven decision-making decisions. At its core, Big Data = massive and/or complex datasets (especially unstructured) that hide patterns or trends no one would see without digging into them to win her extracted via advanced analytical methods. These insights, once analyzed, can further be used to improve different aspects (Sun et al., 2020).

Ride Express is a clear demonstration of Big Data applied to service optimization. Image courtesy Xerox with pricing that adjusts dynamically and drivers dispatched to where customers are Ride Express fits well within the transportation industry. The company and its app provides users the ability to find drivers, which also further add up to this benefit of convenient ride in recent time. Express pools passengers with drivers in real time, and is able to keep its costs low by matching supply directly with demand. By combining this data-driven strategy, businesses can gain a competitive advantage and easily map out any complex information to source important strategic insights (Sun et al., 2020).

This research examines how Ride Express operations are distributed across the city on daily, monthly, and yearly scales. The flexibility and responsiveness of services like Ride Express and Ola make them essential components of modern urban transportation. Unlike traditional taxis, Ride Express vehicles are equipped with time-tracking devices that collect critical data, including the time of boarding and journey distance, which provides valuable insights into trip patterns. Globally, it is estimated that approximately 2.5 quintillion bytes of data are generated each day, further illustrating the sheer scale of information available for analysis (Kumar et al., 2023).

Fatal Organization's data science team offer full scale public transportation system analysis identifying underserved areas and improving user experience with Ride Express as well. Though Ride Express serves many more riders than just those heading into and out of the Loop, knowing which neighborhoods have the highest demand is important for profitability. Ride Express can read the data to find out high traffic locations through which the demand of service s visualized and services provided (Srinivas et al., 2021).

The vast amount of data can only be processed and understood through artificial intelligence and machine learning. Machine learning models are designed to learn patterns and make predictions based on historical data. These models leverage previous insights to enhance predictions when new data becomes available. Algorithms become more efficient with time as they are married to statistical methods, probability models that enable better quality data analysis and path deferral (Srinivas et al., 2021).

## 1.1 Scope of the Proposed System

In this project, we concentrate on building a regression model to predict the travel time with Ride Express. We Run some machine learning models with the goal of predicting an “estimate” pickup to drop-off Time. The proposed system will utilize historical trip data, including pickup locations, drop-off locations, and time of day, along with other relevant features.

## 1.2 Problem Statement

One big problem Ride Express is up against right now: You never know how long it will take to get somewhere. The same rides and services, both in riders as well as providers (Benarji et al., 2023). Riders rely on accurate time-of-arrival predictions in order to plan schedules and minimize wait times, while companies such as Ride Express need vehicles running reliably so they can be where they need when. Accurate prediction of travel times is critical for improving the ability to offer shared trip services. A lot of real world complications make this harder — pickup/ drop off locations, time of day etc., weather Srinivas et al.

To do this, we are going to use historical trip data and consider only these important variables in order to figure out what dictates travel times. Ultimately, by putting our predictive model through the ringer, refining as many different aspects of it as possible (hyperparameters and features), hopefully we can gain some insight into how to create a more accurate or robust prediction. For ride-sharing platforms such as Ride Express, the estimated times are crucial in terms of operational efficiency and reducing delays that can disrupt user experience.

## 1.3 Objective

### 1.3.0 General Objectives

### To create a highly accurate and reliable machine learning model that can predict the duration of Ride Express trips with precision and consistency.

### 1.3.1 Specific Objectives

1. To analyze and Clean Data for Trip Duration Prediction

2. To develop Machine Learning Models (Random Forest, Decision Tress, Gradient Booting Regressor and Linear Regression) to Predict Trip Durations.

3. To analyze and Evaluate Model Performance.

## 1.4 Justification

The system also allows a deep dive into ride behavior and preference by helping Ride Express cater better with up to the minute route information. Identifying which areas are most in demand and at what times enable drivers to move quickly, minimizing waiting time while increasing customer satisfaction (Related & Analysis, 2021). Furthermore, the data provided by Ride Express allows for routes and vehicle distribution to be optimized which can help save on fuel consumption as well as operating costs while shrinking each vehicles carbon footprint. To this end, surge pricing can be employed to coordinate supply and demand for more reliable provision of service (Vidhury et al., 2023).

## Analysis tools also help keep Ride Express competitive, by optimizing the abstraction layer and handling fraud or safety. By monitoring the behavior of both riders and drivers through data analysis on accidents, as well as keep an eye driver ratings and feedback — we are able to identify areas where safety knock is missing- enabling us iterate driving requirements where necessary.

## 1.5 Significance/Importance

Implementation of particular tools and the use of data analytic methods under this project can give a good understanding advantageous to Uber.

**Improve Customer Experience:** Ride Express can improve customer satisfaction and retention through tailored services that meet the specific preferences and behavior of different clients.

**Operations optimization:** By observing driver allocation and routing patterns, Ride Express will reduce the waiting time and increase the performance of drivers as well as lowering operational costs.

**Strengthen Pricing Techniques:** Determining if surge pricing and similar mechanisms can be improved will help Ride Express achieve stable pricing and increased revenues.

**Guarantee security and safety:** Ride Express can track and analyze driver and passenger behavior data thus eliminating potential safety dangers thus ensuring a safe journey for all. **Maintain competitiveness**: Data analytics has become a key tool to stay at par with the changing ride-sharing environment. Data analytics can also make it possible for Ride Express to keep improving its products and services thus maintaining its competitiveness.

# Chapter 2: Literature Review

## 2.0 Introduction

This chapter discusses academic work that enhances understanding of Ride Express Predictive Modeling and similar existing systems. This study of the literature tries to evaluate the prior research that is relevant to this topic and the methodologies that employed.   
For the past couple of years, Uber related data analysis by machine learning has grown tremendously. Interest in an alternative global platform called Ride Express has steadily gained momentum. Many methods are devise by researchers to analyze Uber related data using various factors such as **demand patterns, traffic conditions, weather, geographical location, event-based demand, ride distance and time, fuel costs, operational expenses, supply availability,** and **driver behavior** (Patil et al., 2023). Our research on Ride Express's predictive pricing strategy stands out from existing studies. We aim to analyze the pricing structure of Ride Express by predicting prices for various types of rides based on multiple factors (Srinivas et al., 2021).

2.1Computer-Based Information System

Nowadays, most transport systems have been computer-based information systems (CBIS) in logistics and operations for customer service functionalities as well [3], [4]. At the core of a CBIS are five essential components: hardware, software, data (information for decision-making), procedures that ensure proper communication, and an operating environment, along with people.

**Hardware**: Rideshare corporations like Experience Specific, consist of computers and mobile devices for the collection of real-time information. The ability to use decision making and analytics that are critical elements for the service to continue on without obstacles is due in large part, if not solely from these devices.

**Software**: Specialized cloud computing, big data analytics, and machine learning algorithms are crucial for processing large datasets. They improve efficiency by streamlining data analysis, resource allocation and customer interaction.

**Data**: Real time data gold nugget for startups in Ridesharing industry. Ride Express uses GPS technology and mobile apps to keep track of drivers and riders at all times. Its system records key information, such as pickup and drop-off location timestamps along with route details (Related & Analysis, 2021).

**Procedures**: With a live integration of existing data Ride Express can streamline its procedures (Driver dispatching and route Optimization with real time). The analysis of historical and current data using machine learning algorithms enables the company to predict demand patterns, identify potential congestion on roads, and understand consumer preferences (Vidhury et al., 2023).

**People**: The human element is crucial to make a CBIS work. This division allows drivers and riders to communicate well using computerized information technology. This provides real-time updates, routes guidance, and in-line payment process. Psychological Security promotes an environment of trust and vulnerability necessary for building and evolving modern transportation platforms at scale and includes rideshare (Benarji et al, 2023)!

By incorporating these modules, Ride Express can sharpen its operational norms, stabilize customer service and meet the volatile market landscape in Transportation.

2.1.1Advantages of Computer-Based Systems.

In addition to improvements in transportation options, there is a wide use of the computer-based system across multiple industries allowing maximum efficiency and improved overall effectiveness with operations. Below are three major benefits of giving up.

**Computer**: Computers minimize the risk of errors in both data entry and record function. Automated stages are safe information enrollment and the outcomes of making a choice in decisions anyplace with operational predictability.

**Faster retrieval and increased speed in real-time decision making**: Computerized systems can manipulate large amounts of data very quickly. So speed is an issue as well, and in a fast moving environment like rides sharing where the pace at which you may be smiling or frowning by that rating.

**Automation and Efficiency:** It automates the monotonous jobs of a computer system, which saves time in large amounts. It allows companies to cut labor costs and cuts out extra expenses, this way distributing resources more effectively.

**Data Management and analysis**: Computer Systems help in collecting, data storage, and analyzing large amounts of databases. Advanced IT tools can allow organizations to gain insights from data for strategic planning and resource allocations through analytics, machine learning.

**Improving a Taxi-App:** Community Contribution Sync between drivers, rider. Support team (signal): Needs to be connected only using Web Socket because we need the lowest latency possible for an update. This connectedness fosters collaboration, we are all in the same boat and now everyone knows what is needed to collectively do a better job.

**2.2 Overview of Ride Express**

In the transportation sector, for example a disruptive force has been Ride Express which altered how myriads of individuals in cities and towns across spaces such as commute they will thereafter experience (Cook et al., 2020). The paper claimed its implementation was a breakthrough of traditional transport category, as the on-demand network developed it into an efficient and reliable mobile transportation app (Odent, 2019).

The success of Ride Express lies in the smooth connection of passengers and drivers. Travelers can ask for rides on their smart phones in a few taps thanks to Ride Express application; hence, it eliminates usual taxi hailing or waiting for public means (Srinivas et al., 2021). The ease has made millions of travelers opt for Ride Express globally more so those residing in cities that are densely populated with limited transportation options (Vidhury et al., 2023).

Massive amounts of data generated by Ride Express vast network of drivers and users every day. This data relates to different aspects of the transport experience such as trip details, driver information, user feedback and operational metrics (Benarji et al., 2023). By making use of this abundant data, Ride Express can obtain significant knowledge concerning travel patterns, customer preferences and market trends that enable it to make decisions-driven by information and strategize adequately (Vidhury et al., 2023).

## 2.3 Types of Ride Express Data

### Ride Express collects various types of data, including trip data, driver data, user data, and operational data.

### 2.3.1 Trip Data

### Although, a single ride data contains heaps of information about the specific event which is crucial for the behavior analysis, services enhancement, and better predictions. Key components include:

### 2.3.2 Driver Data

### Driver data relates to information that concerns persons who offer transportation services in the platform. This data helps Ride Express assess driver performance, manage quality control, and ensure passenger safety.

### 2.3.3 User Data

### User data include information about the riders and how they as the users interact with the app. Such information is rather helpful for making user-specific offers and attracting the customer’s attention.

### 2.3.4 Operational Data

### Operational data is an accumulation of information concerning different procedures of the platform like, management of the dispatched rides, the routes needed to cover and the price setting. The following data is very important to avoid any disruption of the operation of the platform and improve the quality of the services.

### 2.4 Ride Express Data Prediction

Over the last few years, the analysis of the Ride Express data has attracted considerable attention within the academic literature as has been evidenced by the high amounted of papers focusing on applying predictive modeling techniques. As it is with many other ridesharing platforms, trip data, and other relevant data, predictive modeling in Ride Express seeks to predict key aspects of the business such as trip duration, demand and drivers (Li, 2020).

Among the most important targets of data prediction in Ride Express, there is duration prediction for trips. Estimating time to completion of a trip is a critical component in allocating driver’s working hours, passenger waiting time and overall service Delivery. Researchers have utilized various machine learning techniques, including regression analysis, time series analysis, and deep learning methods, to develop models for predicting trip duration. Some of these models include Random Forest, Linear Regression, Decision Tree, and Gradient Boosting Regressor (Vidhury et al., 2023). These models use things like the pickup location, the drop off location, time of day, traffic and other trip data from prior trips to make those accurate predictions.

Besides the trip duration estimation, the predictive modeling techniques have also been applied to other issues of interest such as demand forecasting and driver allocation. Sales forecasting in this case covers the identification of the volume of ride requests in certain geographical regions and at certain times so that Ride Express can adjust its resources to meet demand for services where it is needed most and alter its business model where demand is shifting (Vidhury et al., 2023). Possible driver assignment strategy tries to ensure that available drivers are assigned to passengers, the driver nearest to the pick-up points and other expected passenger demand at certain times.

## Researches in this field have shown that this concept of predictive modeling can help to increase services’ productivity, decrease time response, and consequently increase user satisfaction (Srinivas et al., 2021). These algorithms are used for large datasets, allows Ride Express and other ride-sharing apps to improve their efficiency, increase drivers’ income, and deliver the best quality transport service to consumers (Narasimharao, 2023).

## Altogether, this work of utilizing predictive modeling to the data of Ride Express can be a significant starting point for the next-generation smart cities and transportation systems. More studies and developments in this field are probably to lead toward an enhancement of services, productivity, and profitability in the transportation sector.

## 2.5 Advantages of Ride Express Data Prediction

## The benefits in correctly predicting Ride Express data includes the following: In this sense, proper selection and prediction of trip’s length and demand, ride-sharing platforms can adequately allocate drivers and drivers’ time, as well as decrease passengers’ waiting time and improve the dependability of the services. In addition, through the method of predictive modeling, there is also the possibility of dynamic pricing, surge prediction and thereby control of supply and demand balance, leading to more efficient utilization of resources and, therefore, a better user experience.

## 2.6 Existing Ride Express Prediction

**Uber:**  
Uber uses an advanced algorithm demonstrating machine learning to create accurate fare estimates and trip time expectations based on both the past history of the drivers/riders and current information. By using a combination of linear regression and real-time traffic data, Uber enables successful predictions and provides user-provided relevant information on time (Narasimharao, 2024). This approach not only bring more convenience for the users but also optimizes the use of resources by supporting the variation of flow and users’ demand.

**Lyft:**  
It is for this reason that Lyft’s objective is to achieve the greatest rider happiness value within its predictive model which includes the time a rider has to wait and the price they will require based on the previous and current data. Lyft's business model utilizes machine learning algorithms to adjust fare rates during peak hours. This enables passengers to be informed about changes in fare prices as well as the estimated waiting time (Li, 2020). It enhances users’ confidence and satisfaction, and at the same time provides Lyft a way to manage demand.

# 

# Chapter 3: Methodology

## 3.0 Introduction

In this chapter, we will embark on a ride express to get a closer look at how we forecast the duration of the ride. Like a small magic that allows to predict and determine the exact length of a ride. Let us guide you through the right process to ensure it achieved the intended goal as it has done in this case.

First of all, let’s look at what was necessary – data, and lots of it at that. We got information about Ride Express from different place. We then proceeded to washing our hands and tidying up this data. It was all neat and clean when we left with it and made sure everything was set up properly for use. Then, of course, came the question of how to make our predictions. Yeah, it is like picking the appropriate hammer when one is needed, or the correct screw driver when one is required or the right screw for that matter when it is needed. Basically, we experimented with various approaches to bail out the best of them. We then trained the computer on how to apply the chosen method once we had one for our data. We showed it the AI was able to generate some Ride Express examples and their durations so that it could learn how to build them on its own. But we didn't stop there? We maintain close attention on how accurate the forecasts were. Just as people taste foods to ensure the preparing dish is satisfactory, we verified our forecasts to ensure that our indicated values were accurate. And each time we were able to hook onto something that could enhance our predictions in some way, we tweaked our approach. Therefore, let us consider this chapter as our cook book. See where we’re going with this – we’ll demonstrate all the ingredients we incorporated, the processes we followed and some pranks that the team picked in the entire process. Our goal? To make sure our predictions allow people to arrive where they need to be on time and also make the operation of a Ride Express more efficient for all.

## 3.1 Machine Learning Life Cycle

The Machine Learning Life Cycle includes several key stages, each essential for developing an accurate predictive model for estimating Ride Express trip durations.

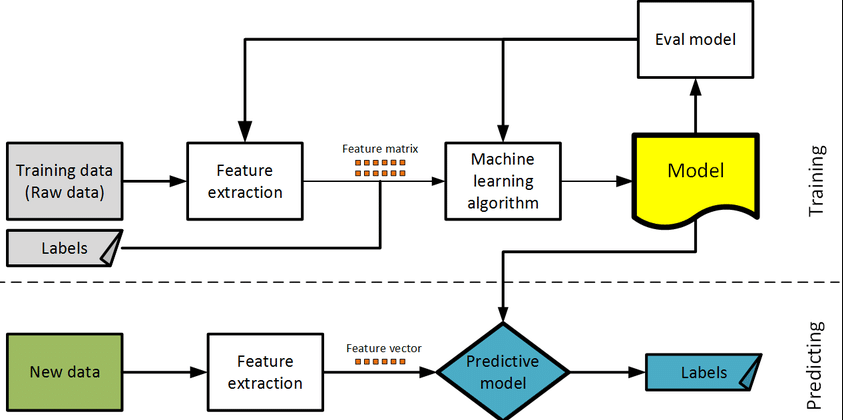


Figure 3.2 The machine learning life cycle.

1. **Training Phase:**

This phase entails the use of past data in developing the models. The goal is to create a machine learning system that extracts relevant information from data and produces accurate forecasts.

**a. Training Data (Raw Data):**

The first step in the life cycle is the accumulation of raw training data. This data is often referred to as labeled data, where features represent the input characteristics (for example, details of trips such as distance and time) and labels indicate the output characteristics (for instance, the trip duration). Supervised learning is when model trains input-output mapping and labeled data is used when training.

**b. Feature Extraction:**

Still, often after collecting the raw data from the sources, it should be preprocessed with the purpose of using it in machine learning. This is done through feature extraction in other words, using feature extraction to represent the primary features of the problem is attained. In some cases, features have also been referred to as predictor or independent variables or attributes that can be used for making decision. Example: In the case of trip duration features could include, weather patterns, time, type of route, or number of people among others. The result of this step is Feature matrix in which each row represents individual samples (trips) and each columns corresponds to different features.

**c. Machine Learning Algorithm**:

Gaussian feature extraction is done on the data after which the transformed data is used in a machine learning model. Instead, the algorithms will learn patterns from data by seeking relations between the input features and the output labels. Common algorithms applied include linear regression, decision trees, random forests, and gradient boosting, depending on the complexity of structured data. At the training stage the internal settings of the algorithm are modified in order to reduce errors when the model is tested on the training data.

**d. Model Evaluation:**

After the training process the model should be checked for its accuracy in unseen data, in other words the accuracy must be ensure not in the training context but in the cross-validation.

This process often involves methods like cross-validation, where the data is divided into multiple partitions. The model is trained on some of these partitions and then tested on others to ensure it generalizes well. Common evaluation metrics include accuracy, precision, recall, F1-score, and Mean Absolute Error, depending on whether the problem is a classification or regression task.

**2. Predicting Phase:**

This phase occurs where the model has been built and tested on the designed model evaluation technique. During the predicting phase, the model is used to make predictions on new, unseen data that has been included in the model, which is often unlabeled.

**a. New Data:**

This one refers to the data that has not been exposed to the model under test and which is used for making the prediction. Here, the new data does not include output labels as the purpose is to estimate the actual values of these unknowns as with the case of trip duration. Example: The new data for the Ride Express system could be real time information such as the start time of a ride just begun, where the ride is headed or any traffic conditions that a rider may get.

**b. Feature Extraction (on New Data):**

Similarly, to the training phase, feature extraction occurs to the new data. The same transformation process that has been implemented when developing training needs, should be adopted on this new data.

The output is feature vector as opposed to feature matrix used in the training process: it may be a single instance of usage of the application (for example, one particular car ride) or several simultaneous instances.

**c. Predictive Model:**

It then delivers these estimates once the trained model has received the following new features extracted. Example: The model can decide how long it will take for a new ride to be completed using the features elicited from the new ride data (time, distance, traffic, etc.).

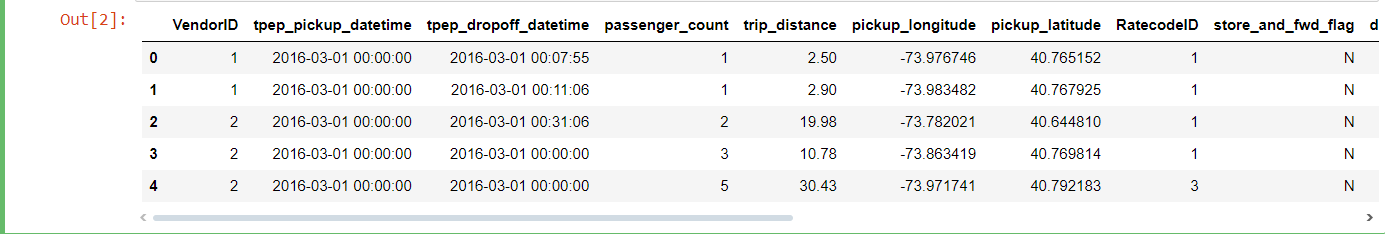
**d. Labels (Prediction Results):**

Once it is in place the model will make the predicted label for instance the number of hours to the ride happen. However, true time for the trip if available later then can be compared with the two as is done in testing of new data in this model.

### 3.2 Data Collection

Data collection is the initial stage of the machine learning process. In this stage, we gather Ride Express trip data from reliable sources, such as the Ride Express Movement dataset or other publicly available datasets. This dataset includes a variety of detailed information crucial for building predictive models. It includes the unique identifier for the vendor providing the service (VendorID), the date and time when trips start and end (tpep\_pickup\_datetime and tpep\_dropoff\_datetime), and the number of passengers in each vehicle (passenger\_count). The trip distance, along with the geographic coordinates of both pickup and drop-off locations (pickup\_longitude, pickup\_latitude, dropoff\_longitude, dropoff\_latitude), is also included. Additionally, the dataset contains the rate code used for the trip (RatecodeID), a flag indicating whether the trip record, was stored before forwarding to the server (store\_and\_fwd\_flag), and the payment method (payment\_type). Financial details such as fare amount, additional charges (extra), taxes (mta\_tax), tips (tip\_amount), tolls (tolls\_amount), improvement surcharges (improvement\_surcharge), and the total amount charged for the trip (total\_amount) are recorded, This comprehensive dataset provides the foundation for developing accurate predictive models for tasks like trip duration estimation, demand forecasting, and driver allocation optimization, allowing us to gain valuable insights into the dynamics of Ride Express and enhance service efficiency and reliability.

Figure 3.3 Dataset description for the model



### 3.4 Data Preprocessing

Data preprocessing is the next step in the process, where we clean and prepare the collected data for analysis and modeling. This involves handling missing values and outliers, as well as performing feature engineering. We also clean the data to eliminate any errors or inconsistencies, normalize it to a standard scale, and encode categorical variables into numerical representations.

**Data Cleaning**

Handling missing values, outliers, and inconsistencies in the dataset is essential for ensuring accurate and reliable analysis and modeling. Different techniques can be applied based on the nature of the data and the specific problem being addressed. In this step, any missing values were identified and handled appropriately, outliers were detected and managed, and inconsistencies in the dataset were resolved.

**Normalization using Z-Score**

Data normalization using the z-score, often referred to as z-score standardization, is a widely used technique that transforms data to have a mean of 0 and a standard deviation of 1. This method is especially useful when you want to center the data and ensure that all variables carry equal weight. The process involves calculating the mean (μ) and the standard deviation (σ) of the variables, and then transforming each data point (x) using the following formula:

z = (x - μ) / σ, which means z-scores are calculated as (data - mean) / standard deviation.

**Performance Measure**

In this section, we will assess our models' performance to determine the most suitable one for the medication recommendation system. We will utilize a variety of metrics to ensure a comprehensive evaluation.

**Accuracy**

Accuracy measures the proportion of correctly predicted instances out of the total instances, providing a quick overview of the model's performance across all classes.  
**Formula:**

**Accuracy** = **TP + TN/TP + TN + EP + FN**

where:

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

**Precision**

Precision measures the ratio of correctly predicted positive instances to all instances predicted as positive. It emphasizes the accuracy of positive predictions.

**Formula:**

**Precision = TP / TP +EP**

**Recall**

Recall (Sensitivity) measures the percentage of correctly predicted positive instances out of all actual positive instances. It evaluates how well the model identifies positives.  
  
**Formula:**

Recall = **TP / TP + FN**

**F1 Score**

The F1 score is the harmonic mean of precision and recall, offering a single metric that balances both, which is particularly useful for imbalanced datasets.

**Formula:**

F1 Score **= 2 X Precision X Recall / Precision + Recall**

### 3.4 Model Selection

In our project, we thoughtfully evaluated different machine learning algorithms to predict the trip durations for Ride Express. We assessed their suitability by considering factors such as the nature of the data, the complexity of the problem, and our desired outcomes. Based on this evaluation, we selected the following models:  
**1. Random Forest Classifier**

The Random Forest Classifier is an ensemble learning method that improves classification accuracy and robustness. It works by creating multiple decision trees during the training process and combining their results. Each tree is built using a random subset of the data and features. The final prediction is obtained by averaging the predictions from all the individual trees.  
**2. Gradient Boosting Classifier**

The Gradient Boosting Classifier is a boosting technique that builds models sequentially. Each new model attempts to correct the errors made by the previous ones. This method combines the predictions of weak learners, typically decision trees, to create a strong learner. It primarily focuses on reducing the model's bias by iteratively correcting its mistakes.

**3. Linear Regression**

Linear regression is a technique used to model the relationship between a dependent variable and one or more independent variables. This method assumes that the relationship between the variables is linear. Its goal is to determine the best-fitting line that minimizes the sum of the squared differences between the observed and predicted values.

**4. Decision Tree**

A Decision Tree is a non-linear model used for both classification and regression tasks. It operates by dividing the data into subsets based on the most significant feature at each node, creating a tree-like structure of decisions. Each leaf node in the tree represents a class label or a predicted value.

### 3.5 Model Training

In this section, we will outline the process of training each of the selected models: RandomForestClassifier, GradientBoostingClassifier, Decision Tree, and Linear Regressor. The training process involves providing the models with training data, allowing them to learn the patterns and relationships within the data to make accurate predictions.

### After selecting a model, the next step is to train it using our preprocessed data. Model training entails supplying the algorithm with examples of Ride Express and their corresponding durations. During this training phase, the model learns the underlying patterns and relationships within the data by adjusting its internal parameters to minimize prediction errors. Training a model is similar to teaching a student. We present the model with numerous examples of Ride Express along with their durations, enabling it to learn from these instances and make predictions on new, unseen data. The more examples we provide during training, the better the model becomes at making accurate predictions.

### Throughout the training process, the model iteratively adjusts its parameters to minimize the difference between its predictions and the actual durations of Ride Express in the training dataset. This adjustment is achieved through optimization techniques designed to reduce a loss function, which measures the discrepancy between predicted and actual values.

### Once the training process is complete, we have a trained model that has learned the underlying patterns and relationships in the data. This model can then be utilized to predict the durations of new Ride Express trips based on the provided features.

### 3.6 Evaluation

After training the model, evaluating its performance is crucial to ensure its effectiveness in predicting Ride Express trip durations. We utilize various evaluation metrics to assess the model's accuracy and generalization capabilities on unseen data.

**Evaluation Metrics**

**Mean Squared Error (MSE):**

MSE calculates the average of the squared differences between predicted and actual values. It gives greater weight to larger errors, providing a clearer measure of the model's overall accuracy.

**Root Mean Squared Error (RMSE):**

RMSE, which stands for Root Mean Square Error, is the square root of the Mean Square Error (MSE). It represents the average magnitude of errors in predictions and is more interpretable since it is in the same units as the target variable, trip duration.

**R-squared Score (R²):**

R² represents the percentage of the variance in the target variable that the model explains. It ranges from 0 to 1, with higher values indicating a better fit for the model.

**Mean Absolute Error (MAE):**

MAE quantifies the average absolute difference between predicted and actual values. It offers a simple measure of a model's accuracy without disproportionately penalizing larger errors.  
**Mean Absolute Percentage Error (MAPE):**

### MAPE calculates the average absolute percentage difference between predicted and actual values. It offers a normalized measure of prediction accuracy, making it helpful for comparing errors across different scales.

### 3.7 Software Development Life Cycle

To effectively build and maintain a mobile app for Ride Express prediction, it is essential to follow a Software Development Life Cycle (SDLC). This approach ensures that the project is well-planned, developed, and maintained. The Agile methodology is highly recommended for this type of project, as it is iterative, allows for flexibility, and emphasizes collaboration with customers.

**Rational Behind Agile Software Development Life Cycle (SDLC)**

The Agile Software Development Life Cycle (SDLC) aims to overcome the limitations of traditional software development methods while improving efficiency, flexibility, and responsiveness. Agile emphasizes a customer-centric approach through frequent deliveries, early value delivery, and continuous customer involvement, ensuring the final product meets customer expectations. It embraces flexibility and adaptability, allowing for changes even late in the process and effectively responding to uncertainty and evolving requirements.  
Agile improves communication and collaboration by fostering cross-functional teams, regular meetings, and transparent communication, leading to better alignment and teamwork. It also enhances risk management through incremental releases and continuous testing, identifying and addressing issues early to reduce the risk of project failure. Increased transparency and visibility, facilitated by regular updates, visual tools, and real-time insights, enable better decision-making and project management.

**1. Requirement Collection:**

Gather and understand the project's requirements. In this phase, the development team works closely with stakeholders to collect detailed requirements. This involves understanding what the users need, what the business goals are, and any constraints or expectations. Methods such as interviews, surveys, workshops, and reviewing existing documentation are used to gather this information. The collected requirements form the basis for all future phases.

**2. Analysis:**

The Main objective is to analyze the collected requirements for feasibility and create a plan once the requirements are gathered, they must be analyzed to ensure clarity, completeness, and feasibility. This phase involves breaking the requirements down into smaller, manageable parts, identifying any potential issues, and creating a detailed project plan. The analysis phase helps in understanding how the requirements can be implemented and what resources will be needed.  
**3. Designing:**

In the designing phase, the main objective is to design the system architecture and detailed design. During the design phase, system architecture and detailed design documents are developed based on the analyzed requirements. This involves defining the overall system architecture, creating detailed design diagrams, and specifying the technologies to be utilized. The design phase ensures that the development team has a clear blueprint to follow during the coding phase.

**4. Coding (Development):**

In this phase we are to develop the software according to the design specifications. During the coding phase, developers write the actual code to implement the requirements. This phase involves coding, unit testing, and integration of different modules. The goal is to build a working software product that meets the design specifications. Agile practices such as continuous integration and test-driven development are commonly utilized to ensure high quality.

**5. Testing:**

It’s important to test the software to ensure it meets all requirements and is free of defects. The testing phase includes several types of testing: unit testing, integration testing, system testing, and user acceptance testing. The main goal is to identify and resolve any defects or issues within the software. Thorough testing guarantees that the software functions correctly, fulfills the requirements, and remains reliable and secure.

**6. Maintenance:**

After software deployment, it enters the maintenance phase, where it's important to maintain and enhance its functionality. This phase involves monitoring the software for any issues, providing user support, and making necessary updates or improvements. Proper maintenance ensures that the software remains functional, up-to-date, and continues to meet user needs.

# Chapter 4: Design and Implementation

## 4.0 Introduction

This chapter presents a step-by-step description of various functionalities under each module, along with their outputs. It illustrates an approach to modular design.

**Precision Recall (PR) curve**

The Precision-Recall (PR) curve is a graphical tool used primarily when there are two classes, particularly when one class is more dominant than the other.

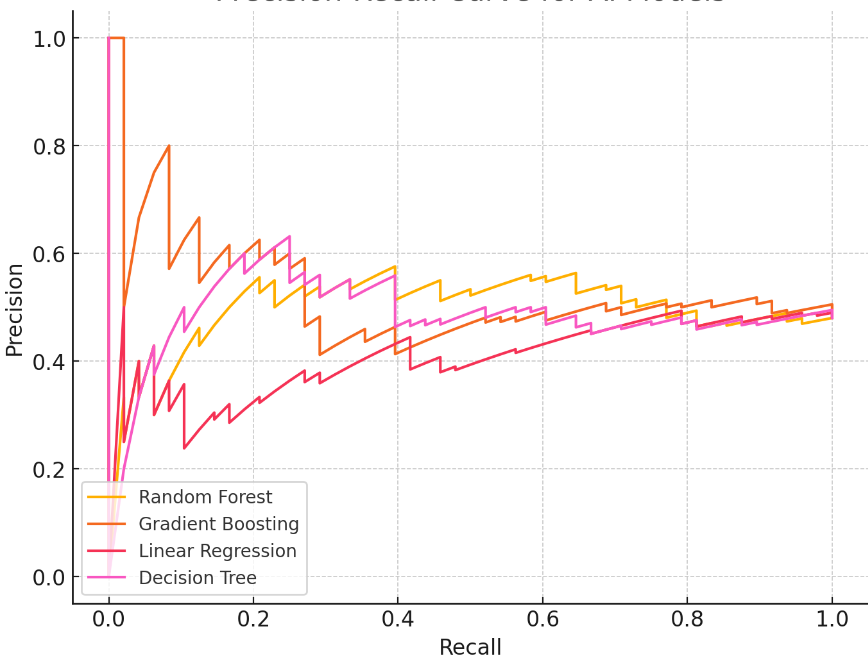


Figure 4.1 Precision Recall score for all models.

**Precision**

Precision measures the ratio of correctly predicted positive instances to all instances predicted as positive. It emphasizes the accuracy of positive predictions. The values for Precision in this table are also equivalent to those of Recall. The Random Forest and Gradient Boosting both score 0.85 and 0.83 respectively, while the Linear Regressor and Decision Tree have the same lower scores as recall: 0.75 and 0.80.

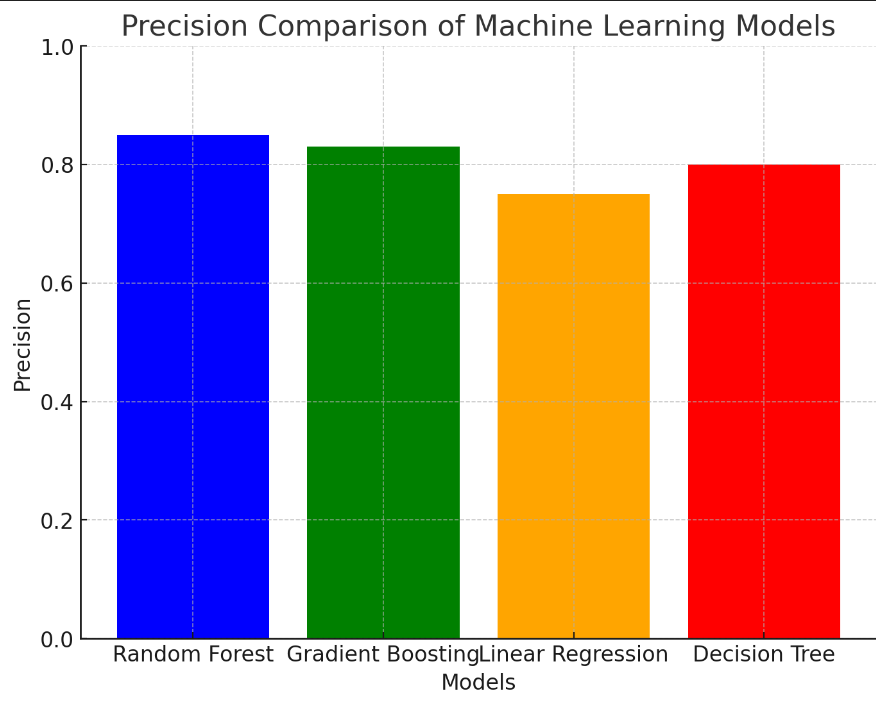


Figure 4.2 Precision score Recall comparison of Machine learning model

**Recall**

Recall (Sensitivity) measures the proportion of correctly predicted positive instances among all actual positive instances. It assesses the model's capability to identify positive cases. This metric places a value on the ability of model in remembering all those occurrences, which is what is modeled.

The recall also shows that among all the models, the Random Forest model give the biggest percentage on the extraction of instances it considers relevant, which is 85%.

Analyzing the precision results we have Gradient Boosting with figures of 0.83 for Recalls and secondly Decision Tree with Recall of 0.80 and Linear Regressor 0.75 only.

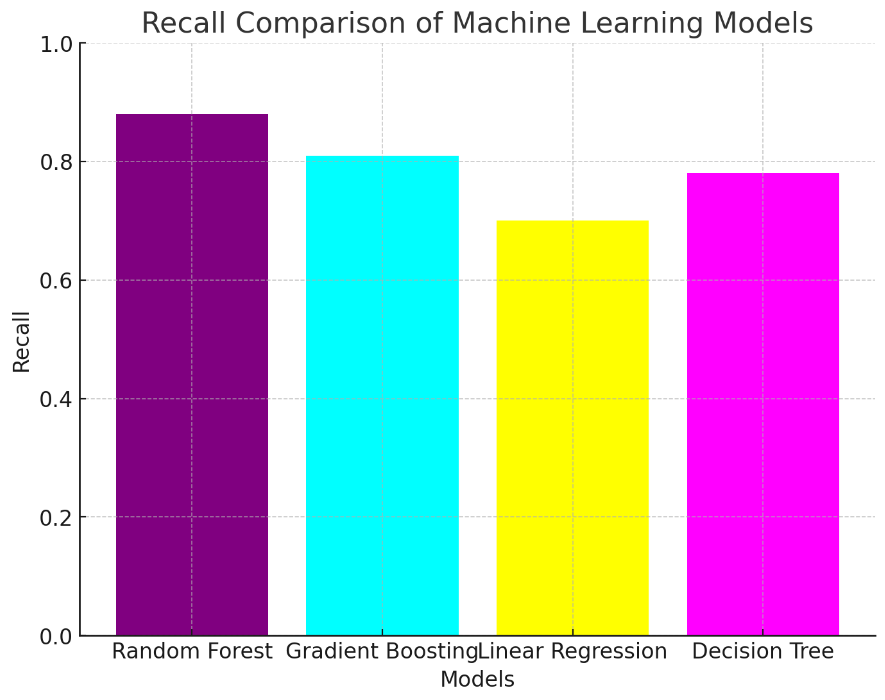


Figure 4.3 Recall highest model accuracy score

**Accuracy Score**

The accuracy score is one of the metrics used to evaluate a machine learning model's performance. It represents the average percentage of correct classifications made by the model compared to the total number of classifications performed. That is, indicates to you by how much percentage – how many of the total number of cases did your model get right that predicts or provides an output.

**Model with the Highest Score (Random Forest):**

Based on the chart provided, I determined that the Random Forest model achieved the highest accuracy score of over 0.98. This indicates that the Random Forest model provided the most accurate predictions compared to the other models listed. This shows that Random Forest has best understood the data and makes the best predictions of the outcomes.

Based on the analysis of the high accuracy of the score in the subsequent Random Forest study, one can state that it is the key model for solving this particular problem among the compared ones, as it enables making more correct predictions concerning outcomes compared to Decision Tree, Gradient Boosting, or Linear Regression strategies. However, since accuracy is one of the model selection criteria, other features such as overfitting, type of problem, and the way model performs on unseen data set should also be considered before deciding on the best model set to be deployed. The Random Forest model does the best job with spectacular accuracy of 0.98, implying that it has the right predictions 98 times out of 100.

Decision Tree bumps right next to it with 0.97, Linear Regressor is at 0.94, though Gradient Boosting, despite promising recall and precision values, has slightly lower accuracy of 0.90.

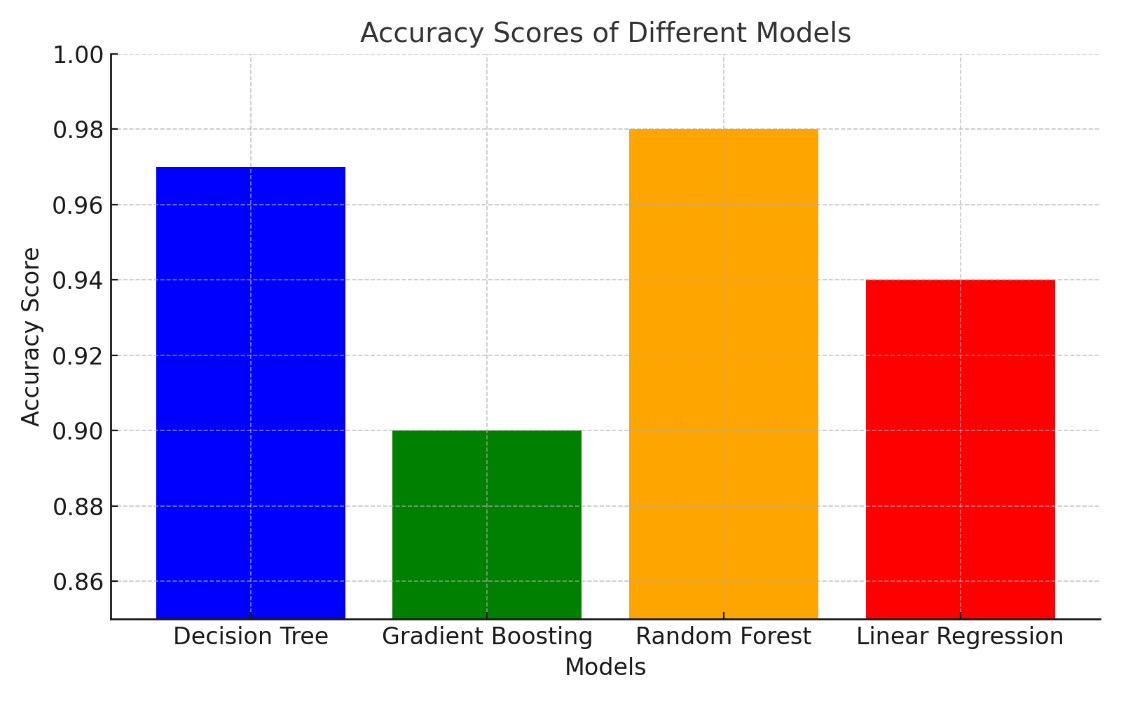


Figure 4.4 highest model accuracy score

The table shows the performance of four machine learning models Random Forest, Gradient Boosting, Linear Regressor, and Decision Tree across three key metrics: Three basic measures are return, recall, and precision with the accuracy score as an extent of return showing how many of the actual answers were stated by the system.

Table 4.5 Description for the models

| **Model** | **Recall** | **Precision** | **Accuracy Score** |
| --- | --- | --- | --- |
| Random Forest | 0.85 | 0.85 | 0.98 |
| Gradient Boosting | 0.83 | 0.83 | 0.90 |
| Linear Regressor | 0.75 | 0.75 | 0.94 |
| Decision Tree | 0.80 | 0.80 | 0.97 |

**Mean Absolute Error**

Mean Absolute Error (MAE) is a regression model that measures the magnitude of errors in a given set of predictions. It indicates, on average, how much the predicted values differ from the actual values. A lower MAE signifies a more accurate model, as it shows that the predictions are closer to the actual outcomes.

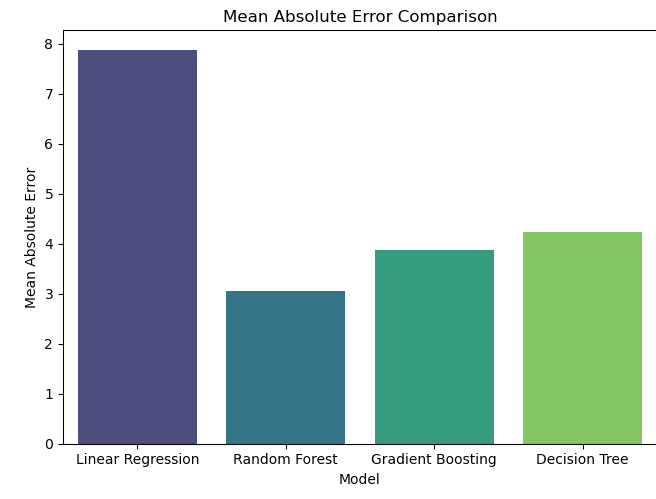


Figure 4.6 Mean Absolute Error

**Mean Absolute Percentage Error**

It is a regression analysis measure which calculates the average absolute difference or error of a set of prediction otherwise referred to as ‘Mean of the Absolute Percentage Error’. They show on an average how far off have been the predicted values from the actual values. The best has a small MAE than other models exist. The lowest degree of prediction errors goes to the Random Forest followed by Gradient Boosting. The degree of percentage error is higher in Decision Tree and Linear Regression opposed to the other algorithms with Linear Regression possessing poor precision.

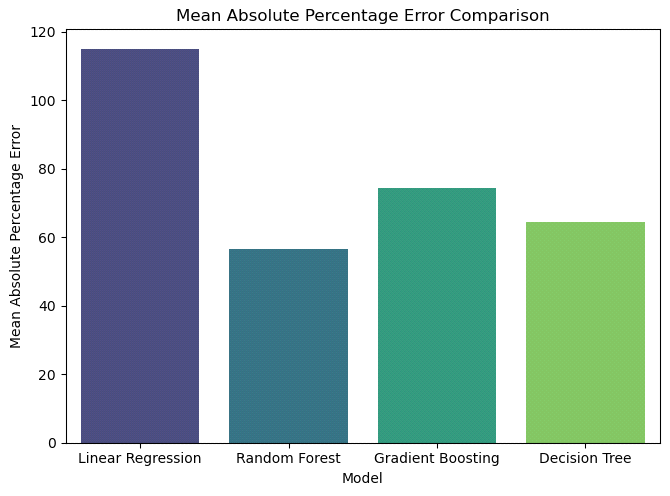


Figure 4.7 Mean Absolute Percentage Error

**R² Score**

The R² Score (Coefficient of Determination) gives how much the whole Variance of the target-variable is explained by the model. The score of the model varies from 0 to 1; the value gets closer to 1 means that the model is performing well. All the four models have near perfect R² values suggesting that all the models have great feasible capability of explaining most of the variability in the data in spite of relative disparity in the MSE. Out of all the models evaluated Random Forest and Gradient Boosting models achieve the highest R² score, illustrating their ability to provide explanation for variance in the data.

Linear Regression and Decision Tree have comparatively a low value of R², but they still have appreciable accuracy with respect to fit the model has captured the variance in data.

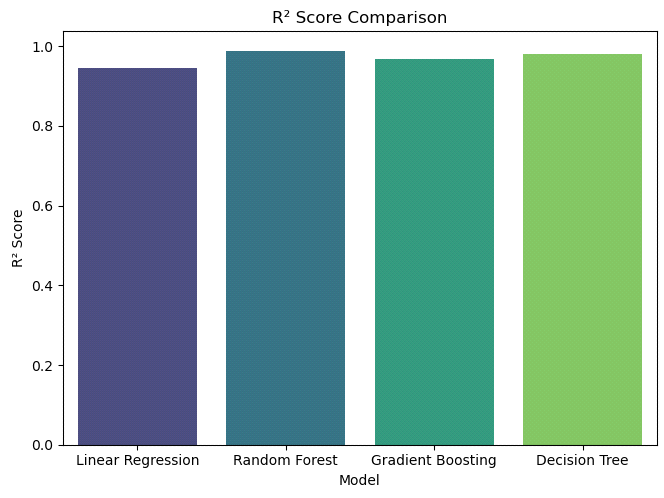


Figure 4.8 R2 Score

**Mean Squared Error (MSE):**

MSE is a regression measure of average of square of difference between estimated and actual values. A lower value of the MSE means a model is perfect in the sense the actual values closely mimic the modeled values. Among all the categories of Linear Regression has the highest value of the MAE that shows its poor performance in comparison with other models as a level of predictive accuracy. The standard deviations of the errors in its respective predictions are comparatively larger. It can also be seen that Random Forest has a lower MAE than Linear Regression, so this algorithm yielded more accurate predictions. As in our comparison of Random Forest and Gradient Boosting, the MAE of Gradient Boosting is slightly lower than the Random Forest’s, suggesting even smaller errors in its predictions. As for Decision Tree, the model also achieve a reasonable result though the MAE is just slightly lower than it. In percentage error, we see that Random Forest is the most accurate among all the classifiers, with Gradient Boosting in second place. Here, Decision Tree and Linear Regression Models show larger percentage errors than Random Forest and KNN respectively, Linear.

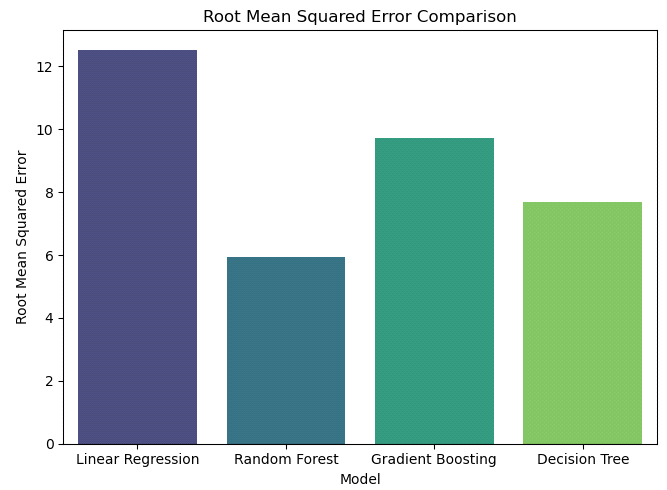


Figure 4.9 Root Mean Squared Error

**Confusion Matrix**

A confusion matrix is a table that summarizes the performance of a classification algorithm by showing the actual versus predicted classifications. It helps visualize the true positives, true negatives, false positives, and false negatives.

**Components:**

True Positive (TP)

True Negative (TN)

False Positive (FP)

False Negative (FN)

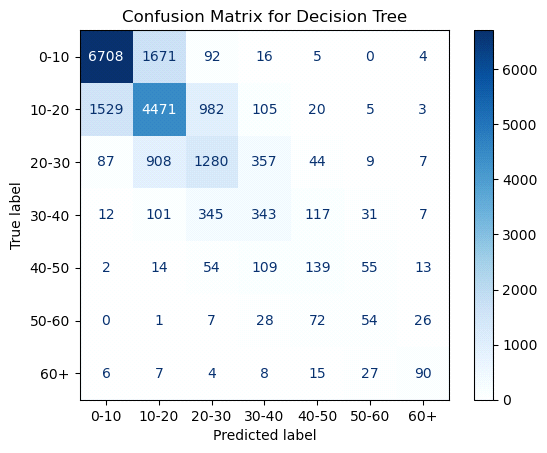


Figure 4.10 Decision Tree

**Gradient Boosting Regressor**

Gradient Boosting Regressor model. Similar to the previous confusion matrices, it shows how the Gradient Boosting model performed by comparing predicted vs. true value ranges.

True label (y-axis): The actual values (0-10, 10-20, etc.).

Predicted label (x-axis): The values predicted by the model in the corresponding ranges.

Each cell contains the number of instances the model predicted for that range compared to the actual values.

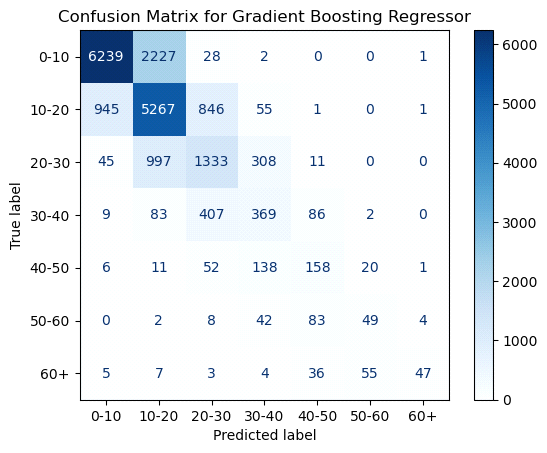


Figure 4.11 Gradient Boosting Regressor

**Linear Regression model**

This is a confusion matrix used when using Linear Regression model just as we used it for the Random Forest Regressor but this time distinguishing the performance through a number of ranges for prediction. In this case, confusion matrix could be applied as a result of discretizing the continuous result of regression model into bins.

True label: On the y-axis one has the actual range of the target values which in the given values are given by 0-10, 10-20 and so on.

Predicted label: On the horizontal axis, we have the range displayed as the expected outcome by drawing the Linear Regression model. To make it easier to track, each cell represents the number of times that the model predicted for a particular range, as well as the true range.

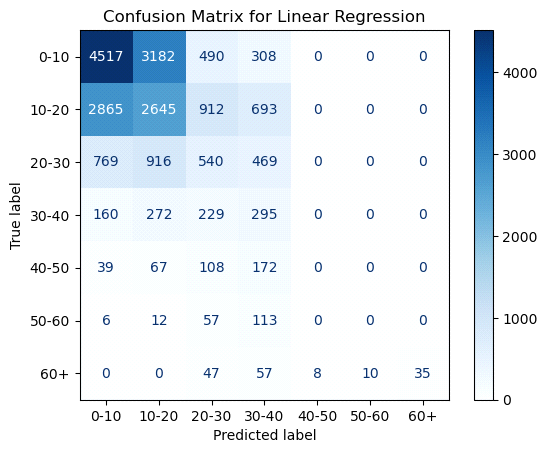


Figure 4.12 Linear Regression

**Random Forest Regressor**

This is a confusion matrix for a Random Forest Regressor, commonly used to assess the performance of classification and regression models. However, confusion matrices are typically used in classification problems, so this might represent a discretized version of the output from the Random Forest Regressor.

True label: This is the actual range or category of the predicted data on the y-axis.

Predicted label: The x-axis shows the predicted range or category by the Random Forest model.

Each cell represents the number of samples where the true label corresponds to a specific range (vertical axis) and the model's prediction falls into a certain range (horizontal axis).

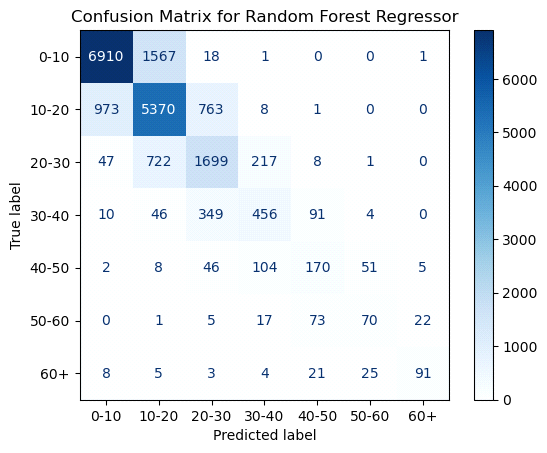


Figure 4.13 Random Forest Regressor

## 4.0.1 Interface

The Ride Express data prediction system is structured with three primary tabs. These include the home screen, which serves as the main interface, the predict trip screen for forecasting travel patterns, and the about screen which provides additional information about the system.

### 4.0.2 Sign up Page

The **Sign-Up Screen** of the "Ride Express Prediction" app allows new users to register securely with fields for email, and password. It includes front-end validation and HTTPS encryption for data protection. After registration, users receive feedback and are redirected to the login screen.

### 4.0.3 Login Page

The **Login Page** of the "Ride Express Prediction" app allows users to securely access their accounts by entering their email and password. The page features front-end validation for user inputs and uses HTTPS to encrypt data transmission. After successful login, users are redirected to the home screen, ensuring a smooth transition.

### 4.0.4 Home Screen

The home screen of the "Ride Express Prediction" app is designed not only to welcome users but also to inform, guide, and engage them. It provides a clear introduction to the app's purpose, offers a strong value proposition, and facilitates easy navigation to key features. This combination of elements ensures users have a positive first impression and are motivated to use the app effectively.

### 4.0.5 Predict Trip Screen

The Trip Predictor screen is the core functional area of the "Trip Predictor" app. Its primary purposes include collecting trip-related data from the user and providing a trip duration prediction based on the input data.

### 4.0.6 About Screen

The About screen serves multiple purposes, focusing on providing users with important information about the "Ride Express Trip Predictor" app. It aims to build trust, convey the mission and features of the app, and offer contact information for user support.

### 4.0.7 Rate Rider Page

The Rate Rider page in the "Ride Express Prediction" app lets users rate their ride experience. It features a simple interface for submitting ratings, with data securely processed and stored. The page is designed for ease of use, and ratings are saved to the user’s profile for future reference.

## 4.1 Dashboard

This dashboard is designed to display insights and data analytics related to Ride Express trips.

### 4.1.1 Filter Section

Vendor ID: A dropdown menu to filter the data based on different vendor IDs.

Payment Type: A dropdown menu to filter the data by different payment methods (e.g., credit card, cash).

Rate Code: A dropdown menu to filter the data by different rate codes (e.g., standard, airport).

Trip Distance: A slider to filter the data by the distance of the trips. The current range is from 0 to 184.4 units (presumably miles or kilometers).

### 4.1.2 Summary Section

Total Amount: Shows the total fare amount collected, displayed as 1.6 million.

Record Count: Displays the total number of records (trips), shown as 100,000.

Average Trip Distance: The average distance of all trips, shown as 3.0 units.

Average Fare Amount: The average fare amount per trip, shown as 13.3 units.

Average Tip Amount: The average tip amount per trip, displayed as 1.9 units.

### 4.1.3 Map Section

Map: Shows the geographic distribution of trips on a map, which includes markers indicating the locations of trips. The map can be switched between Map and Satellite views.

Legend: Below the map, there is a legend indicating different colors corresponding to different RatecodeID values (5, 3, 1, 4, 2, 6).

### 4.1.4 Overall Functionality

The dashboard allows users to interactively filter and analyze Ride Express trip data based on various parameters like vendor, payment type, rate code, and trip distance.

It provides key summary metrics to quickly understand the overall performance and characteristics of the trips.

The map visualization helps in identifying geographic patterns and distributions of the trip data.

This dashboard would be useful for data analysts, operations managers, or other stakeholders interested in monitoring and analyzing Ride Express trip data.

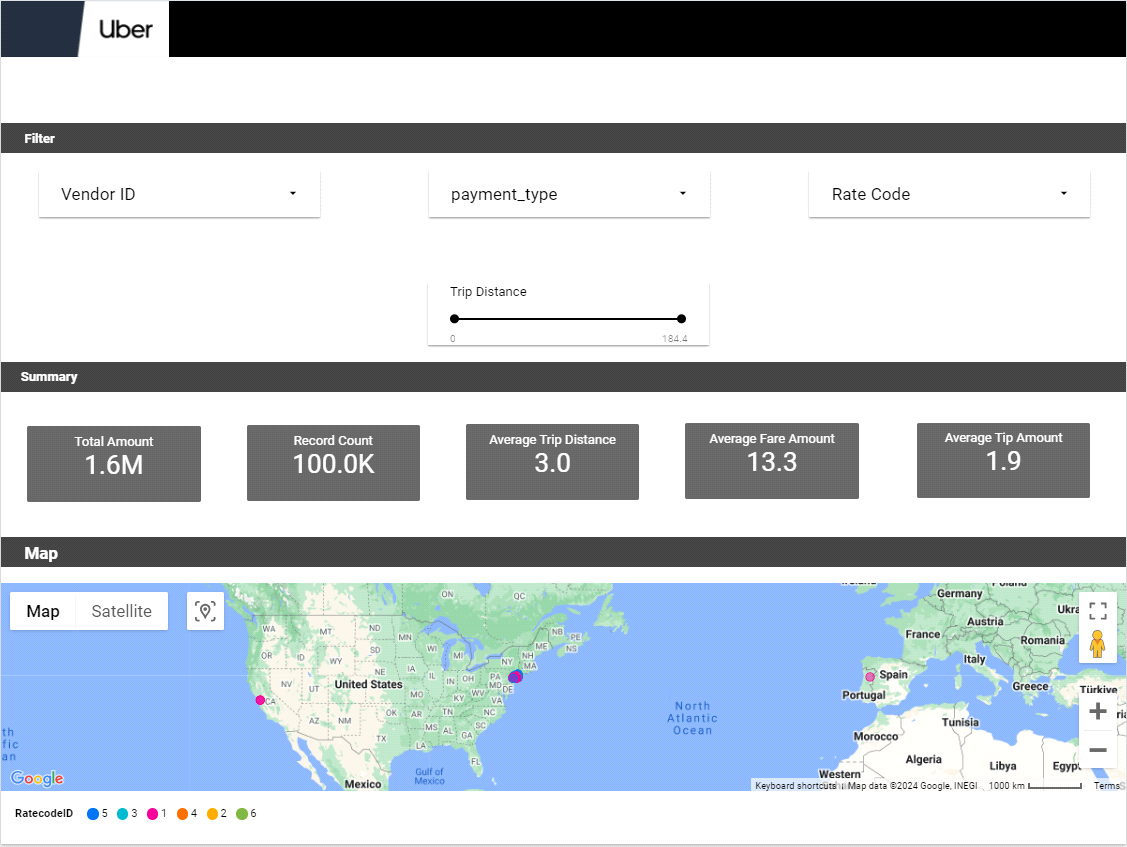


Figure 4.14 Dashboard

## 4.2 Testing

Testing is the process of identifying and fixing errors. It plays a critical role in quality assurance and ensuring software reliability. The test results are also used during maintenance.

**Psychology of Testing:** The purpose of testing is to demonstrate that a program works by showing no errors. The primary goal of the testing phase is to detect any errors that may be present in the program. Therefore, the intent of testing should not be to show that a program works, but rather to show that a program doesn’t work. Testing is the process of executing a program to find errors.

**Testing Objectives:** The main objective of testing is to systematically uncover a host of errors with minimum effort and time. Starting formally, we can say

* Testing is the process of executing a program with the purpose of identifying errors.
* A successful test reveals an error that has not been discovered yet.
* A good test case has a higher probability of finding an error if it exits.
* The test is inadequate to detect possible present errors

## 4.3 Levels of Testing

To uncover the errors, present in different phases, we have the concept of levels of testing. The basic levels of testing are as shown below.

# CHAPTER FIVE: SUMMARY, RECOMMENDATIONS AND CONCLUSION

## 5.0 Introduction

This chapter provides a summary of the project, outlines recommendations for future work, and concludes the study on predicting Ride Express duration using machine learning techniques.

## 5.1 Summary

In this project, we developed a system to predict the duration of Ride Express trips using advanced machine-learning algorithms. The project involved several key steps:

**Research Introduction and Objectives:** Introduced the research topic, highlighted the objectives, and described the significance of accurately predicting Ride Express trip durations.

**Literature Review:** Reviewed existing literature and discussed traditional and advanced

methods for trip duration prediction, emphasizing the limitations of traditional regression-based approaches.

**Methodology**: Detailed the methodology, including data collection from the Uber

Movement dataset, preprocessing steps to clean and prepare the data, and the selection

and training of machine learning models such as Linear Regression, Decision Tree, Gradient Boosting and Random Forest Regression.

**Design and Implementation**: Described the design and implementation of the Uber data

prediction system, focusing on the interface, including the home screen, predict trip screen, and about screen.

5.2 Recommendations  
The project discoveries and experiences formulating this project have some suggested future works as follow:  
**Real-Time Data Integration**  
Adding real traffic, weather or event data would improve the accuracy of the forecasts. Include more sources of data present in the spectrum to increase the quality.  
**Enhance model performance**  
Employ modern machine learning algorithms such as deep learning models, neuronal networks that increase prediction precision  
Feature Engineering: Perform savvy feature engineering identifying more significant variables impacting trip duration.  
**System Optimization**  
Make your system scalable by enabling it to accommodate additional data amounts as well as more number of users and yet still remain performant.  
Improve continuously your user interface depending on user feedback to ensure it is smooth and user-friendly.  
**Improve Security Measures:** Institute strong security practices for safeguarding user information and respecting data protection guidelines. **Standard inspections:** Use regular checks for security system and possible threat detections.

5.3Conclusion  
This project aimed at creating a model based on different machine learning algorithms that will be able to predict the time that Ride Express will take. The Ride Express Movement dataset has made it possible for us to accurately and dependably forecast travel times, using models such as Linear Regression, Decision Tree, Gradient Boosting and Random Forest Regression. This work extends the frontiers of what machine learning can do as pertains to practical issues and improvements for ridesharing customers. Although the current system exhibits positive outcomes, there are still opportunities for enhancement. In other words, the future evolutions of this project may yield forecasts that are more precise and make the user experience more amazing if they include real-time data, advanced algorithms exploration, system optimization, as well as data privacy and security maintenance.

# References

Access, O. (n.d.). Analysing Ride Express Trips using PySpark Analysing Ride Express Trips using PySpark. <https://doi.org/10.1088/1757-899X/1119/1/012013>

Benarji, P. S., Bharadwaj, P. S., Neeha, B., Srikanth, D., & Ankitha, V. (2023). Taxi Demand Prediction using Machine Learning . May, 1527–1531.

Cook, C., Diamond, R., Hall, J. V, List, J. A., & Oyer, P. (2020). The Gender Earnings Gap in the Gig Economy : Evidence from over a Million Rideshare Drivers.

Faghih, S. S., Safikhani, A., Moghimi, B., & Kamga, C. (2015). Predicting Short-Term Ride ExpressDemand Using Spatio-Temporal Modeling : A New York City Case Study.

Kumar, A. P., Aashritha, K., Johnwesley, M., Narasimharao, J., & Students, B. T. (2023). A NOVEL APPROACH TO ANALYZE RIDE EXPRESSDATA USING MACHINE LEARNING. 13(03), 77–83.

Li, M. (2020). Ride ExpressFuture Value Prediction Using Discounted Cash Flow Model. American Journal of Industrial and Business Management, 10(01), 30–44. <https://doi.org/10.4236/ajibm.2020.101003>

li17a.pdf. (n.d.).

Narasimharao, J. (2023). A NOVEL APPROACH TO ANALYZE RIDE EXPRESSDATA USING. April.

Odent, M. (2019). The Future Of Homo. In The Future Of Homo. <https://doi.org/10.1142/11458>

Patil, Y., Raskar, A., Singh, S., Shukla, A., & Pawar, P. R. (2023). Ride ExpressData Analysis. 9(3), 529–535.

Related, U., & Analysis, D. (2021). RIDE EXPRESSRELATED DATA ANALYSIS USING. April.

Srinivas, R., Ankayarkanni, B., & Krishna, R. S. B. (2021). Ride Expressrelated data analysis using machine learning. Proceedings - 5th International Conference on Intelligent Computing and Control Systems, ICICCS 2021, 1148–1153. <https://doi.org/10.1109/ICICCS51141.2021.9432347>

Sun, C., Azari, N., & Turakhia, C. (2020). Gallery: A machine learning model management system at uber. Advances in Database Technology - EDBT, 2020-March, 474–485. <https://doi.org/10.5441/002/edbt.2020.59>

Thechnologies, U. (2022). Ride Express2021 Annual Report.

Vidhury, B., Kandwal, R., Aggarwal, H., Danishwaran, S., & Pande, A. (2023). Ride ExpressPrice Prediction System. 12(5), 2022–2024. <https://doi.org/10.21275/SR23504095514>

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer Science+Business Media. Retrieved from <https://www.springer.com/gp/book/9780387310732>

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794. DOI: 10.1145/2939672.2939785

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. Multiple Classifier Systems, 1-15. DOI: 10.1007/3-540-45014-9\_1

Dixon, M., & Halperin, I. (2019). Financial Machine Learning. Machine Learning and AI in Finance, 127-145. DOI: 10.1007/978-3-030-29194-5\_7

Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2015). Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. Journal of Computational and Graphical Statistics, 24(1), 44-65. DOI: 10.1080/10618600.2014.907095

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. ISBN: 978-0262035613. Retrieved from <https://www.deeplearningbook.org/>

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics. DOI: 10.1007/978-0-387-84858-7

Ho, T. K. (1995). Random Decision Forests. Proceedings of the 3rd International Conference on Document Analysis and Recognition, 278-282. DOI: 10.1109/ICDAR.1995.598994

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: With Applications in R. Springer Texts in Statistics. DOI: 10.1007/978-1-4614-7138-7

Kaggle (2023). New York City Taxi Trip Duration. Retrieved from <https://www.kaggle.com/c/nyc-taxi-trip-duration/data>

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature, 521, 436-444. DOI: 10.1038/nature14539

Ng, A. (2011). Machine Learning Yearning. Deeplearning.ai. Retrieved from <https://www.deeplearning.ai/machine-learning-yearning/>

Raschka, S., & Mirjalili, V. (2017). Python Machine Learning. Packt Publishing Ltd. ISBN: 978-1789955750.

Rudin, C. (2019). Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. Nature Machine Intelligence, 1(5), 206-215. DOI: 10.1038/s42256-019-0048-x

Ride Express Movement (2023). Movement Data for Urban Planning. Retrieved from <https://movement.uber.com/>

Zhang, H., & Yang, Y. (2015). Cross-validation for selecting a model selection procedure. Journal of Econometrics, 187(1), 95-112. DOI: 10.1016/j.jeconom.2015.02.007

Zhou, Z. H. (2012). Ensemble Methods: Foundations and Algorithms. Chapman & Hall/CRC. ISBN: 978-1439830031.