A Project report on

##### Transportation Demand Prediction

##### using Machine Learning

Submitted in partial fulfillment of the requirements for award of the degree of

**BACHELOR OF TECHNOLOGY**

In

**Computer Science & Engineering**

By

**Ella Kumar Swamy** (21U41A0510)

|  |  |  |
| --- | --- | --- |
| **Kona Anand** | **Dosuru Sravanthi** | **Villuri Sasi Vardhan** |
| **(21U41A0523)** | **(22U45A0501)** | **(22U45A0513)** |

Under the Esteemed Guidance of Mrs. T. Santhoshi Lakshmi

**Assistant Professor, Department of Computer Science & Engineering**

****

**DEPARTMENT OF COMPUTER & ENGINEERING**

**DADI INSTITUTE OF ENGINEERING &TECHNOLOGY**

**(AN AUTONOMOUS INSTITUTE)**

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to JNTU GV)

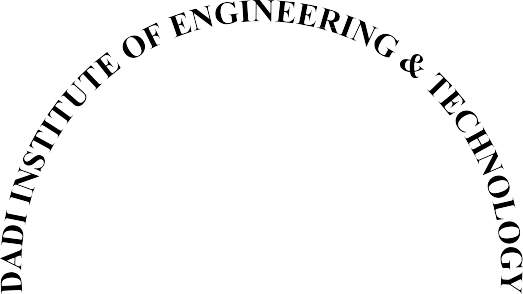
**Accredited by NAAC with ‘A’ Grade and Inclusion u/s 2(f) & 12(B) of UGC Act**

An ISO 9001:2015, ISO 14001:2015 & ISO 45001:2018 Certified Institute.

**NH-16, Anakapalle – 531002, Visakhapatnam, A.P.**

**(2021-2025)**





**DEPARTMENT OF**

**COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that the project report entitled **“TRANSPORTATION DEMAND PREDICTION USING MACHINE LEARNING”** submitted by E. Kumar Swamy (21U41A0510), K. Anand (21U41A0523), D. Sravanthi (22U45A0501), V. Sasi Vardhan (22U45A0513). In partial fulfilment of the requirements for award of the Degree of **Bachelor of Technology** in **Computer Science and Engineering**, from Dadi Institute of Engineering & Technology(A), Anakapalle affiliated to JNTUGV, accredited by NAAC with 'A' grade is a record of bonafide work carried out by them under my guidance and supervision.

**PROJECT GUIDE HEAD OF THE DEPARTMENT**

**EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We would like to express our gratitude to our project guide, **Mrs. T. Santhoshi Lakshmi**, **Assistant Professor** of the **Computer Science & Engineering**, for her guidance and assistance throughout the development of this project. Without her supervision, support, and encouragement, we would not have gained awareness of many new things during our project.

We convey our heartfelt thanks to **Dr. K. Sujatha**, **Professor & Head of the Department of Computer Science & Engineering**, for motivating us to successfully complete the project.

We would like to thank **Principal Dr. R. Vaikunta Rao**, Dadi Institute of Engineering & Technology, for providing the necessary facilities to carry out our project work successfully.

We express our gratitude to the Teaching and Non-teaching Staff of the Department of **Computer Science & Engineering**, who have been directly and indirectly involved in this journey, for their encouragement in completing our project.

We would like express our deep sense of gratitude to **Honorable Chairman, Sri Dadi Ratnakar**, of Dadi Institute of Engineering & Technology, for providing necessary facilities to carry out our project work successfully.

Endeavors over a long period can also be successful through constant effort and encouragement. We wish to take this opportunity to express our deep gratitude to all the people who have extended their cooperation in various ways during our project. It is our pleasure to acknowledge the help of all those respected individuals.

**PROJECT ASSOCIATES**

**E. Kumar Swamy (21U41A0510)**

**K. Anand (21U41A0523)**

**D. Sravanthi (22U45A0501)**

**V. Sasi Vardhan (22U45A0513)**

**DECLARATION**

We hereby declare that the project entitled **“TRANSPORTATION DEMAND PREDICTION”** is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology in **Computer Science & Engineering** under esteemed supervision of **Mrs. T. Santhoshi Lakshmi, Assistant Professor**. Thisis a record of work carried out by us and results embodied in this project report have not been submitted to any other university for the award of any Degree.

**PROJECT ASSOCIATES**

**E. Kumar Swamy (21U41A0510)**

**K. Anand (21U41A0523)**

**D. Sravanthi (22U45A0501)**

**V.Sasi Vardhan (22U45A0513)**

##### ABSTRACT

**Transportation demand prediction** is a critical component for effective urban planning and infrastructure management. Accurate forecasting of transportation needs allows for optimized resource allocation, improved service delivery, and enhanced sustainability . DA4PT (**Data Analytics for Public Transport**), for discovering the factors that influence travelers in booking and purchasing bus tickets and analyzing transportation data to inform planning decisions, Transportation Demand Prediction uses machine learning and statistical modeling, to predict future transportation demand. Random Forest Regression model is used to examine correlations between factors and ticket sales using historical booking data and also offers a reliable foundation for forecasting ticket sales by MAE,MSE,RMSPE and by various statistical metrics . Deploying the model with Random Forest Regressor with the parameters with 3 fold cross validation. I found that for training data there is a R2 score of **61.11**% which means the model has predicted well on the datapoints.This leads to low bias. Also the model has a R2 score of **61.54**%. The variance in model is also low. The ticket booking platform can enhance revenue by utilizing these insights to optimize methods and generate additional revenue streams by incorporating these data into their business plan.

**KEYWORDS**: Transportation, Travel Time Analysis , Random Forest Regression , Revenue Sales Production.

**TABLE OF CONTENTS**

1. INTRODUCTION
2. LITERATURE REVIEW
3. EXISTING SYSTEM
4. PROPOSED SYSTEM
5. SYSTEM REQUIREMENTS
6. TECHNOLOGY USED
7. SYSTEM DESIGN
8. SYSTEM IMPLEMENTATION
9. RESULTS
10. CONCLUSION
11. REFERENCES
12. APPENDIX
13. RESEARCH PAPER

### **LIST OF CONTENTS**

ABSTRACT

LIST OF FIGURES

LIST OF ABBREVIATIONS

1. [**INTRODUCTION** 12](#_dfbgy19cxbji)
   1. [BACK GROUND 13](#_xkk6rkks8lnh)

1.2 [MOTIVATION 13](#_ugnswnbtd1oy)

* 1. [PROBLEM DEFINITION 14](#_ju6f2wr58iou)

1.4 [OBJECTIVE OF THE PROJECT 14](#_gqild04832wg)

* 1. [LIMITATIONS OF THE PROJECT 14](#_8cx3knuvu71p)
  2. [ORGANIZATION OF THE PROJECT 15](#_pv2gsvgvhzhk)

1.7 OBJECTIVES 15

* 1. SIGNIFICANCE 15

**2**  [**LITERATURE REVIEW** 17](#_9lq9c8g1031u)

* 1. [Traditional Approaches 18](#_xm1es4b0h5ds)
     1. [Time Series Analysis 18](#_sjettsueos3g)
     2. [Basic Statistical Methods 18](#_cohez64zwj26)
  2. [Machine Learning Techniques 18](#_4cyrv4kc4a9i)
     1. [Regression Models 18](#_4x84y8xmkpk7)
  3. Machine Learning Algorithms 18
  4. [Industry-Specific Applications 19](#_omkorni7y9b4)
     1. [Airline Industry 19](#_umb0dnroicb4)
     2. [Railway Industry 19](#_andbot9tfrq)
     3. [Entertainment Industry 19](#_xyfzedu03abc)
     4. [Conclusion 19](#_kg3owrel9rsg)

1. [EXISTING SYSTEM 20](#_p4ct9l783rgp)
2. [PROPOSED SYSTEM 22](#_d9ol8nd5e7b)
3. [SYSTEM REQUIREMENTS 24](#_2jn1d3vr45p)
   1. [Hardware Requirements 25](#_v5zgq0usf891)
      1. [Minimum Requirements 25](#_h30q9ce4arue)
      2. [Recommended Requirements 25](#_3hnv6hoaaa96)
   2. [Software Requirements 25](#_18glj8yg0ttv)
      1. Operating Systems 25
      2. [Development Tools 25](#_egk7kfgh0505)
      3. Database 25
      4. Web Server 26
      5. [Version control 26](#_l3xzo3m3hpng)
      6. [Other Tools 26](#_27mlk8woxwcy)
   3. [Network Requirements 26](#_z6khjb5krojc)
   4. [Conclusion 26](#_991011bs0ixu)
   5. [Module Wise Functional Requirements 26](#_8q83wa9bjgy2)
      1. [Data Collection and Management Module 26](#_3tsqtyo74zel)
      2. Data Processing Module 27
      3. [Predictive Modeling Module 27](#_1euovypmlapd)
      4. [Prediction and Analysis Module 28](#_1k47xah5jqoa)
      5. Web Interface 28
      6. [Deployment and Maintenance Module 29](#_8vcsx32cbva6)
   6. [Conclusion 29](#_o2zs5ehcvgq2)
      1. [Performance 30](#_e78nxbaknm1i)
      2. [Reliability 30](#_otbpgdwfogme)
      3. [Usability 30](#_kj4l7flehmbc)
      4. [Security 31](#_hv6uurivjzdq)
      5. [Maintainability 31](#_v7ll95jxe3jo)
4. TECHNOLOGY USED 32
   1. Technologies 32

|  |  |  |
| --- | --- | --- |
| 6.1.1 | Python | 33 |
| 6.1.2 | Pandas | 33 |
| 6.1.3 | Numpy | 33 |
| 6.1.4 | Scikit-learn | 34 |
| 6.1.5 | Matplotlib | 34 |
| 6.1.6 | Seaborn | 35 |

1. SYSTEM DESIGN 36
   1. Architecture for the “Transport Demand Prediction” 37
   2. [Workflow 38](#_gf4f2zyqannv)
   3. [Data Science 38](#_p7t4lasv73zb)
   4. [Machine Learning 39](#_a1009kc06x2l)
   5. [Predictive Analytics 39](#_d4yqzggc2bg0)
   6. [Web Development 39](#_qox8ffogsnrv)
   7. [Software Engineering 40](#_4zf8lzqn1vmc)
2. [System Implementation 46](#_9uisxhkr30wc)
   1. [Data Collection and Preprocessing 47](#_revk8s4cbtn0)
   2. [Feature Selection and Engineering 47](#_3igwthhdd83)
   3. [Model Development 48](#_msti5khpdk7n)
   4. [Model Evaluation and Optimization 48](#_yhcbpy210jvn)
   5. [Prediction and Analysis 49](#_1zri5e3qpdt0)
   6. [User Interface Development 49](#_qsfz431jbj1i)
   7. [Expected Outcomes 49](#_4o74dixd1fxl)
   8. [Conclusion 50](#_7c6ywhuovqoi)
   9. [Algorithms Explanation 50](#_8c6j2bqnzmad)
      1. [Linear Regression 50](#_gjk4vky84jdl)
      2. [Polynomial Regression 50](#_z6dvishpqkd9)
      3. [Ridge Regression 51](#_55f5cvu50n0r)
      4. [Lasso Regression 51](#_vyx09nyircw0)
      5. [Elastic Net Regression 52](#_6vvqpvn1m9m7)
      6. [Decision Trees 52](#_t8uhwoj0v54b)
      7. [Random Forests 53](#_y1u4782o6lbl)
   10. [Testing 53](#_ndcnslia8fcn)
       1. Types of Testing 53
          1. Unit Testing 53
          2. Integration Testing 54
          3. Functional Testing 54
          4. Performance Testing 54
          5. Usability Testing 54
          6. Regression Testing 54
          7. Acceptance of Testing 55
       2. Test Cases for the Project 55
3. [RESULTS 57](#_bxe3qrgrgiqb)
   1. Results 58
      1. User Interface 59
   2. [Output 1 61](#_4s6p9x7xnhye)
   3. [Output 2 62](#_jdbtuh7oe6ua)
4. [CONCLUSION 64](#_d8sofuslyd31)
   1. [Future Scope 65](#_hfi4v93pnaoh)
5. [REFERENCES 68](#_iicxpyb4mzg6)
6. APPENDIX 68

[Appendix 69](#_4nrb56is836)

### LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Fig. No.** | **Description** | **Page No.** |
| 3.1 | Performance Evaluation | 20 |
| 7.1 | Architecture | 37 |
| 7.2 | Use case Diagram | 40 |
| 7.3 | Class Diagram | 41 |
| 7.4 | Activity Diagram | 44 |
| 7.5 | Sequence Diagram | 55 |
| 9.1 | Results | 60 |
| 9.2 | Output 1 | 63 |
| 9.3 | Output 2 | 64 |

LIST OF SYMBOLS, ABBREVIATIONS

DA4PT Data Analytical Public Transport MAE Mean Absolute Error

MSE Mean Squared Error

ARIMA Auto Regressive Integrated Moving Average SVM Support Vector Machine

LASSO Least Absolute Shrinkage and Selection Operator RANDOM CV Random forest Cross Validation

XGBOOST Extreme Gradient Boosting

# CHAPTER 1 INTRODUCTION

##### Introduction:

The project "Transportation Demand Prediction" addresses the growing need for efficient and intelligent ticket management systems across various industries, such as airlines, railways, and entertainment. Ticket booking trends are influenced by numerous factors, including seasonal variations, special events, pricing strategies, and customer behavior. Traditional methods of managing ticket allocation often fall short in accurately predicting demand, leading to either overbooking or underutilization of resources. This project leverages the power of regression analysis to develop a predictive model that optimizes ticket booking and enhances overall operational efficiency.

* 1. **Background**

Ticket booking systems have evolved significantly over the years, transitioning from manual processes to sophisticated digital platforms. Despite these advancements, many systems still rely on basic statistical methods and heuristic approaches that lack the precision required for accurate demand forecasting. The advent of big data and machine learning presents an opportunity to revolutionize ticket booking by applying advanced regression techniques to predict future trends based on historical data.

The "Transportation Demand Prediction" project aims to develop a predictive system that forecasts future ticket bookings based on historical data. The system leverages regression analysis to predict demand trends, providing valuable insights for optimizing booking strategies. By analyzing past booking patterns and external factors, the project intends to enhance booking accuracy and resource allocation, ultimately leading to improved customer satisfaction and operational efficiency.

The project encompasses several components, including data collection and preprocessing, predictive modeling, and a user-friendly web interface. The integration of these components aims to create a cohesive system that delivers reliable and actionable predictions to users.

* 1. **Motivation**

The motivation for this project stems from the challenges faced by businesses and organizations in accurately forecasting ticket bookings. In the competitive travel and entertainment industries, precise demand predictions are crucial for optimizing inventory, pricing strategies, and resource allocation. Traditional methods often fall short in accuracy and efficiency, leading to overbooking or underbooking issues.

By implementing regression techniques, the project aims to address these challenges and provide a data-driven approach to booking predictions. This not only improves operational efficiency but also enhances the overall customer experience by reducing discrepancies between actual and predicted bookings.

* 1. **Problem Definition**

The primary problem addressed by this project is the lack of accurate and reliable forecasting for ticket bookings. Current systems may rely on simplistic or outdated methods that fail to account for complex patterns and trends in booking data. This can lead to inefficiencies in inventory management, revenue optimization, and customer satisfaction.

Specifically, the problems include:

* + - **Inaccurate Predictions**: Existing methods may not capture the intricacies of booking trends, leading to unreliable forecasts.
    - **Inefficient Resource Allocation**: Poor predictions can result in either overbooking or underbooking, affecting operational efficiency and customer satisfaction.
    - **Limited Data Utilization**: Traditional systems may not fully leverage historical data or external factors that influence booking patterns.
  1. **Objective of the Project**

The main objectives of the "Transportation Demand Prediction" project are:

1. **Develop a Predictive Model**: Create a regression-based model that accurately forecasts future ticket bookings based on historical data.
2. **Optimize Booking Accuracy**: Enhance the accuracy of booking predictions to improve resource allocation and reduce booking discrepancies.
3. **Provide User-Friendly Interface**: Design an intuitive web interface that allows users to easily input data, view predictions, and access insights.
4. **Analyze Trends**: Offer tools for trend analysis and visualization to help users understand booking patterns and make informed decisions.
   1. **Limitations of the Project**

While the project aims to provide valuable predictions and insights, it is subject to certain limitations:

1. **Data Quality**: The accuracy of predictions depends on the quality and completeness of historical data. Inaccurate or incomplete data can affect model performance.
2. **Model Limitations**: Regression models may not capture all complexities of booking patterns, especially in highly volatile or unpredictable environments.
3. **External Factors**: The model may not fully account for external factors (e.g., economic conditions, special events) that influence booking trends.
4. **Scalability**: Handling large volumes of data and maintaining performance may pose challenges as the system scales.
   1. **Organization of the Project**

The project is organized into the following key phases:

1. **Requirement Analysis**: Identifying functional and non-functional requirements, and defining the scope of the project.
2. **System Design**: Developing the system architecture, selecting technologies, and designing the database schema and user interface.
3. **Data Collection and Preprocessing**: Gathering historical booking data, cleaning, and preprocessing the data for use in predictive modeling.
4. **Model Development**: Implementing and training regression models, and evaluating their performance.
5. **Web Interface Development**: Designing and developing a web-based interface for user interaction and visualization of predictions.
6. **Testing**: Conducting various types of testing (e.g., unit, integration, performance) to ensure the system meets the specified requirements.
7. **Deployment**: Deploying the system on a web server and making it available for users.
8. **Maintenance**: Monitoring the system, addressing any issues, and performing updates as needed.

##### Objectives

The primary objective of this project is to design and implement a regression-based model that can predict ticket booking patterns with high accuracy. The specific goals include:

1. **Data Analysis**: Collecting and analyzing historical ticket booking data to understand the underlying patterns and factors influencing demand.
2. **Model Development**: Implementing various regression models, including linear regression, polynomial regression, and advanced techniques like Ridge, Lasso, and Elastic Net regression.
3. **Model Evaluation**: Assessing the performance of different models using appropriate metrics and selecting the best-performing model.
4. **Prediction and Optimization**: Utilizing the selected model to make accurate predictions and provide actionable insights for ticket allocation and pricing strategies.
5. **User Interface**: Developing a user-friendly interface that allows stakeholders to interact with the model, input data, and visualize predictions.

**Significance**

Accurate demand forecasting is crucial for the success of any ticket booking system. Overbooking can lead to customer dissatisfaction and potential revenue loss, while

underbooking results in wasted resources and missed revenue opportunities. By applying regression analysis, this project aims to enhance the precision of demand forecasts, enabling better resource management and improved customer satisfaction. The insights gained from this model can help in:

* + **Dynamic Pricing**: Adjusting ticket prices based on predicted demand to maximize revenue.
  + **Capacity Planning**: Allocating resources more efficiently to meet demand during peak periods.
  + **Marketing Strategies**: Designing targeted marketing campaigns based on predicted booking trends.

# CHAPTER 2 LITERATURE REVIEW

##### Literature Review

The literature review for the "Transportation Demand Prediction" project examines previous research and methodologies applied in the field of demand forecasting and ticket management. This review covers traditional approaches, advanced machine learning techniques, and specific applications within industries such as airlines, railways, and entertainment.

* 1. **Traditional Approaches**
     1. **Time Series Analysis**

Time series analysis has been widely used for forecasting ticket demand. Techniques like ARIMA (AutoRegressive Integrated Moving Average) have been commonly employed to analyze and predict future bookings based on historical data. Studies by Box and Jenkins (1976) highlighted the effectiveness of ARIMA in capturing seasonal patterns and trends in time series data.

* + 1. **Basic Statistical Methods**

Simple linear regression and moving average methods have also been used for ticket demand forecasting. While these methods are easy to implement, they often fail to capture complex patterns and interactions between multiple influencing factors.

* 1. **Machine Learning Techniques**
     1. **Regression Models**

The application of various regression models has gained popularity in recent years. Linear regression, polynomial regression, and ridge regression have shown promise in predicting ticket demand by considering multiple features simultaneously.

* + - * **Linear Regression**: Studies by Draper and Smith (1998) demonstrated the use of linear regression for predicting airline ticket sales, showing that it can provide reasonable accuracy when the relationship between variables is linear.
      * **Polynomial Regression**: Research by Aiken and West (1991) indicated that polynomial regression could capture more complex relationships, although it requires careful tuning to avoid overfitting.
      * **Ridge and Lasso Regression**: Techniques like Ridge and Lasso regression, introduced by Hoerl and Kennard (1970) and Tibshirani (1996) respectively, address multicollinearity and feature selection, providing more robust models for prediction.
    1. **Machine Learning Algorithms**

Advanced machine learning algorithms, such as decision trees, random forests, and support vector machines (SVM), have also been explored for ticket demand forecasting.

* + - * **Decision Trees and Random Forests**: Studies by Breiman (2001) highlighted the effectiveness of ensemble methods like random forests in improving prediction accuracy by combining multiple decision trees.
      * **Support Vector Machines (SVM)**: Research by Vapnik (1995) demonstrated that SVMs could provide high accuracy in classification and regression tasks, making them suitable for predicting complex booking patterns.
  1. **Industry-Specific Applications**
     1. **Airline Industry**

The airline industry has extensively researched demand forecasting due to the high impact of accurate predictions on revenue management. Studies by Weatherford and Bodily (1992) emphasized the importance of dynamic pricing and demand forecasting in optimizing airline ticket sales. Recent research by Tsui et al. (2017) explored the use of machine learning models, including regression and neural networks, to predict airline ticket demand, showing significant improvements over traditional methods.

* + 1. **Railway Industry**

In the railway sector, demand forecasting is crucial for scheduling and resource allocation. Research by Cadarso et al. (2013) explored the use of regression models to predict train occupancy rates, highlighting the potential for improved operational efficiency. Advanced models incorporating passenger behavior and external factors have been shown to enhance prediction accuracy.

* + 1. **Entertainment Industry**

Ticket sales for events and entertainment have also been studied, with research focusing on predicting demand for concerts, sports events, and movies. Studies by Borghesi (2008) and Courty (2003) explored dynamic pricing strategies and the use of machine learning models to forecast event attendance, demonstrating the potential for maximizing revenue through better demand predictions.

* + 1. **Conclusion**

The literature review reveals a broad spectrum of methodologies and applications for ticket demand forecasting, ranging from traditional statistical methods to advanced machine learning techniques. Regression models, in particular, have shown significant potential in capturing complex patterns and improving prediction accuracy. By leveraging these insights, the "Transportation Demand Prediction" project aims to develop a robust and efficient system for predicting ticket demand, optimizing resource allocation, and enhancing overall operational efficiency across various industries.

# CHAPTER 3 EXISTING SYSTEM

1. **EXISTING SYSTEM**

DA4PT (Data Analytics for Public Transport) is a widely used model in transportation planning. DA4PT, for discovering the factors that influence travellers in booking and purchasing bus tickets. DA4PT focuses on analyzing transportation data to inform planning decisions. DA4PT is considered an existing model for transportation demand prediction due to its reliance on data-driven approaches, proven machine learning algorithms, real-time data integration, and successful implementations in real-world scenarios.

**Models Used in existing systems**:

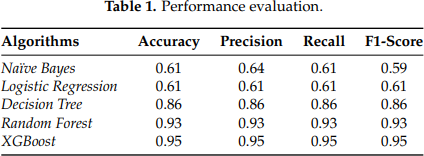


Fig: 3.1 Performance evaluation

**Drawbacks of DA4PT:**

1. Limited to small data
2. Less accuracy
3. Not adaptable
4. Less suitable for long term prediction Features that are extracted from DA4PT:
   * Historical Precedent and Adoption in Industry
   * Data-Driven Nature
   * Advanced Predictive Analytics and Machine Learning Models
   * Optimizing Public Transport Systems
   * Cost and Resource Efficiency
   * Ongoing Research and Improvement

# CHAPTER 4 PROPOSED SYSTEM

1. **PROPOSED SYSTEM**

Proposed system for transportation demand prediction would likely involve a multi- faceted approach, combining various data sources, machine learning models, and real- time updates. Here's a breakdown of the key components and considerations:

**Data Collection:** Gather data on transportation modes, routes, volumes, and patterns from mobiticket App.

**Data Analysis:** Examine data to identify trends, patterns, and correlations.

**Route Choice Modeling:** Study how people choose routes.

**Market Segmentation:** Divide the market into segments based on demographics, travel behavior, and other factors.

**Travel Demand Modeling:** Use models to forecast travel demand based on factors like population growth, land use, and transportation infrastructure.

**Validation:** Model cross-validation is a technique used to evaluate the performance of a machine learning model by training and testing it on multiple subsets of the data. This helps to ensure that the model is not overfitting or underfitting and provides a more accurate estimate of its performance.

By conducting a system study for transportation demand prediction, planners and policymakers can make informed decisions about infrastructure investments, transportation services, and policies to manage demand and reduce congestion

# CHAPTER 5

**SYSTEM REQUIREMENTS**

1. **System Requirements**
   1. **Hardware Requirements**
      1. **Minimum Requirements**:
         * **Processor**: Intel Core i3 or equivalent
         * **RAM**: 4 GB
         * **Storage**: 10 GB of free space
         * **Graphics Card**: Integrated graphics or basic GPU
      2. **Recommended Requirements**:
         * **Processor**: Intel Core i5 or equivalent
         * **RAM**: 8 GB or more
         * **Storage**: 20 GB of free space (SSD preferred)
         * **Graphics Card**: Dedicated GPU (e.g., NVIDIA GeForce GTX 1050 or higher) if working with large datasets or complex models
   2. **Software Requirements**
      1. **Operating System**:
         * **Windows**: Windows 10 or later
         * **macOS**: macOS 10.15 (Catalina) or later
         * **Linux**: Ubuntu 18.04 LTS or later (or other compatible distributions)
      2. **Development Tools**:
         * **Python**: Version 3.7 or later (recommended Python 3.9 or later)
         * **IDE/Editor**:
           + PyCharm (Professional or Community Edition)
           + VS Code
           + Jupyter Notebook
         * **Libraries/Packages**:
           + pandas: For data manipulation and analysis
           + numpy: For numerical operations
           + scikit-learn: For machine learning algorithms and evaluation metrics
           + matplotlib and seaborn: For data visualization
           + xgboost or lightgbm (optional): For advanced gradient boosting techniques
           + flask or django (if developing a web-based user interface)
      3. **Database** (if applicable):
         * **Relational Databases**:
           + MySQL 5.7 or later
           + PostgreSQL 12 or later
         * **NoSQL Databases** (if needed):
           + MongoDB 4.2 or later
           + SQLite (for lightweight applications)
      4. **Web Server** (if developing a web interface):
         * **Flask** or **Django**: For developing web applications
         * **Apache** or **Nginx**: For serving web applications (optional, depending on deployment needs)
      5. **Version Control**:
         * **Git**: For version control and collaboration
         * **GitHub** or **GitLab**: For hosting repositories and collaboration
      6. **Other Tools**:
         * **Virtual Environment**: venv or conda for managing Python dependencies
         * **Data Handling**:
           + Excel or CSV file handling libraries (e.g., pandas)
   3. **Network Requirements**

* **Internet Access**: Required for downloading libraries, accessing datasets, and deploying web-based applications.
* **Firewall Settings**: Ensure that necessary ports are open if deploying a web application or database server.
  1. **Conclusion**

The system requirements listed above ensure that the development environment is adequately equipped to handle the data processing, model training, and user interface components of the "Transportation Demand Prediction" project. By meeting these requirements, you can ensure smooth development, testing, and deployment of your ticket booking prediction system.

* 1. **Module wise functional requirements**
     1. **Data Collection and Management Module**

**Objective**: To collect, store, and manage historical ticket booking data.

**Functional Requirements**:

1. **Data Collection**:
   * **Input**: Ability to ingest data from various sources (e.g., CSV files, databases).
   * **Formats**: Support for multiple data formats (e.g., CSV, Excel).
2. **Data Storage**:
   * **Database**: Store collected data in a structured format within a relational database (e.g., MySQL, PostgreSQL).
   * **Backup**: Implement mechanisms for regular backups and data integrity checks.
3. **Data Management**:
   * **CRUD Operations**: Provide Create, Read, Update, and Delete functionalities for managing ticket booking records.
   * **Data Cleaning**: Identify and handle missing, erroneous, or duplicate data entries.
4. **Data Access**:
   * **Query Interface**: Allow for querying and retrieving data based on specific criteria (e.g., date range, booking class).
     1. **Data Preprocessing Module**

**Objective**: To preprocess the collected data for use in predictive modeling.

**Functional Requirements**:

1. **Data Cleaning**:
   * **Handling Missing Values**: Implement strategies for filling or removing missing data.
   * **Outlier Detection**: Identify and handle outliers in the dataset.
2. **Feature Engineering**:
   * **Feature Extraction**: Create new features from existing data to improve model performance (e.g., booking trends, seasonal effects).
   * **Normalization/Standardization**: Scale features to ensure they are within a comparable range.
3. **Data Transformation**:
   * **Encoding**: Convert categorical variables into numerical format (e.g., one-hot encoding).
   * **Aggregation**: Aggregate data as needed (e.g., daily, weekly summaries).
4. **Data Splitting**:
   * **Training and Test Sets**: Split data into training and test sets to evaluate model performance.
     1. **Predictive Modeling Module**

**Objective**: To build and train regression models to predict future ticket bookings.

**Functional Requirements**:

1. **Model Selection**:
   * **Algorithm Choice**: Implement various regression algorithms (e.g., Linear Regression, Polynomial Regression, Ridge Regression).
   * **Hyperparameter Tuning**: Optimize model parameters to improve performance.
2. **Model Training**:
   * **Training Process**: Train selected models on historical booking data.
   * **Validation**: Validate models using cross-validation techniques.
3. **Model Evaluation**:
   * **Performance Metrics**: Evaluate models using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared.
   * **Error Analysis**: Analyze prediction errors to refine model performance.
4. **Model Saving and Loading**:
   * **Persistence**: Save trained models for future use.
   * **Loading**: Load saved models for prediction without retraining.
     1. **Prediction and Analysis Module**

**Objective**: To generate predictions and analyze booking trends based on the trained models.

**Functional Requirements**:

1. **Prediction**:
   * **Input Handling**: Accept user input or system-generated data for prediction.
   * **Output**: Generate and display predicted ticket booking numbers.
2. **Trend Analysis**:
   * **Visualization**: Provide charts and graphs to visualize booking trends and prediction results.
   * **Comparative Analysis**: Compare predictions with historical data to assess accuracy.
3. **Reporting**:
   * **Generate Reports**: Create detailed reports of predictions and model performance.
   * **Export Options**: Allow users to export reports in various formats (e.g., PDF, Excel).
     1. **Web Interface Module**

**Objective**: To provide a user-friendly web interface for interacting with the ticket booking prediction system.

**Functional Requirements**:

1. **User Interface**:
   * **Input Forms**: Provide forms for users to input data for prediction.
   * **Result Display**: Display prediction results and visualizations clearly.
2. **Navigation**:
   * **Menus and Links**: Implement intuitive navigation menus for accessing different sections (e.g., prediction input, results, historical data).
3. **User Interaction**:
   * **Feedback Forms**: Allow users to provide feedback or contact support.
   * **Error Handling**: Display user-friendly error messages for invalid inputs or system issues.
4. **Authentication and Authorization**:
   * **User Accounts**: Manage user accounts and permissions (e.g., Admin, End User).
   * **Security**: Implement security measures to protect user data and access.
     1. **Deployment and Maintenance Module**

**Objective**: To deploy the application and ensure its ongoing maintenance and availability.

**Functional Requirements**:

1. **Deployment**:
   * **Hosting**: Deploy the web application on a web server (e.g., Apache, Nginx).
   * **Scalability**: Ensure the system can handle varying loads and scale as needed.
2. **Monitoring**:
   * **Performance Monitoring**: Track system performance and availability.
   * **Error Logging**: Log errors and issues for troubleshooting and maintenance.
3. **Updates and Maintenance**:
   * **Software Updates**: Regularly update the system to fix bugs and add new features.
   * **Data Backup**: Implement regular backups of the application and database.
4. **User Support**:
   * **Help Documentation**: Provide documentation and FAQs for users.
   * **Support Channels**: Offer support channels for users to get help (e.g., email, chat).
   1. **Conclusion**

The functional requirements for each module of the "Transportation Demand Prediction" project provide a clear outline of what each component of the system needs to achieve. By defining these requirements, you can ensure that each module meets its objectives and integrates seamlessly with the overall system.

* + 1. **Performance**

**Objective**: Ensure the system performs efficiently under expected loads.

**Requirements**:

1. **Response Time**:
   * **Prediction Requests**: The system should generate predictions within a specified time frame (e.g., under 5 seconds for prediction results).
   * **Page Load Time**: Web pages should load within 3 seconds to ensure a smooth user experience.
2. **Throughput**:
   * **Concurrent Users**: The system should handle a specified number of concurrent users (e.g., 100 simultaneous users) without performance degradation.
3. **Scalability**:
   * **Load Handling**: The system should be able to scale horizontally or vertically to accommodate increased load and growing data volumes.
     1. **Reliability**

**Objective**: Ensure the system is consistently available and reliable.

**Requirements**:

1. **Uptime**:
   * **Availability**: The system should have a high uptime (e.g., 99.9% uptime) to ensure availability for users.
2. **Fault Tolerance**:
   * **Error Handling**: Implement mechanisms to handle and recover from system errors without significant impact on user experience.
3. **Backup and Recovery**:
   * **Data Backup**: Regularly back up data to prevent loss and ensure recovery in case of failure.
   * **Disaster Recovery**: Implement a disaster recovery plan to restore functionality in case of catastrophic failures.
     1. **Usability**

**Objective**: Ensure the system is user-friendly and accessible.

**Requirements**:

1. **User Interface**:
   * **Ease of Use**: The web interface should be intuitive and easy to navigate for users of all technical levels.
   * **Accessibility**: Ensure the interface complies with accessibility standards (e.g., WCAG) to accommodate users with disabilities.
2. **Documentation**:
   * **User Guides**: Provide comprehensive user documentation and help guides.
   * **Support**: Offer responsive customer support and FAQs to assist users.
     1. **Security**

**Objective**: Protect the system and user data from unauthorized access and breaches.

**Requirements**:

1. **Authentication**:
   * **User Accounts**: Implement secure authentication mechanisms for user accounts (e.g., password hashing, multi-factor authentication).
2. **Authorization**:
   * **Access Control**: Ensure users can only access features and data they are authorized to use.
3. **Data Protection**:
   * **Encryption**: Use encryption for data transmission (e.g., SSL/TLS) and data storage to protect sensitive information.
4. **Vulnerability Management**:
   * **Security Updates**: Regularly update software and libraries to patch vulnerabilities.
     1. **Maintainability**

**Objective**: Ensure the system is easy to maintain and update.

**Requirements**:

1. **Code Quality**:
   * **Standards**: Adhere to coding standards and best practices to ensure code readability and maintainability.

# CHAPTER 6 TECHNOLOGY USED

#### TECHNOLOGIES USED

* 1. **Python**

**Overview**: Python is a high-level, interpreted programming language known for its readability, simplicity, and versatility. It is widely used in data science and machine learning due to its extensive libraries and community support.

**Key Features**:

* **Readability**: Python's syntax is clear and easy to read.
* **Versatility**: Supports multiple programming paradigms (procedural, object- oriented, functional).
* **Extensive Libraries**: Libraries such as pandas, numpy, scikit-learn, matplotlib, and

seaborn are extensively used in data analysis and machine learning.

**Usage in Project**:

* Data cleaning, manipulation, and analysis.
* Implementation of regression algorithms.
* Building and evaluating models.
* Creating visualizations.
  1. **pandas**

**Overview**: pandas is a powerful Python library for data manipulation and analysis. It provides data structures like DataFrames and Series, which are essential for handling structured data.

**Key Features**:

* **DataFrames**: Two-dimensional, size-mutable, and potentially heterogeneous tabular data structure.
* **Data Manipulation**: Functions for filtering, grouping, and aggregating data.
* **Data I/O**: Support for reading/writing data from/to various formats (CSV, Excel, SQL, etc.).

**Usage in Project**:

* Loading and preprocessing ticket booking data.
* Data cleaning and transformation.
* Feature engineering.
  1. **numpy**

**Overview**: numpy is a fundamental package for numerical computing in Python. It provides support for large multi-dimensional arrays and matrices, along with a collection of mathematical functions.

**Key Features**:

* **Array Operations**: Efficient operations on arrays and matrices.
* **Mathematical Functions**: Functions for linear algebra, statistics, and other mathematical operations.
* **Performance**: Optimized performance for large-scale numerical computations.

**Usage in Project**:

* Numerical operations on data.
* Handling arrays and matrices required for machine learning algorithms.
  1. **scikit-learn**

**Overview**: scikit-learn is a robust machine learning library for Python that provides simple and efficient tools for data mining and data analysis. It supports various machine learning algorithms, including regression, classification, clustering, and more.

**Key Features**:

* **Machine Learning Algorithms**: Implementations of various algorithms (e.g., linear regression, decision trees, random forests).
* **Model Evaluation**: Metrics and tools for evaluating model performance (e.g., MAE, MSE, R-squared).
* **Preprocessing**: Tools for scaling, encoding, and transforming data.

**Usage in Project**:

* Implementing and training regression models.
* Evaluating model performance.
* Performing data preprocessing and feature selection.
  1. **matplotlib**

**Overview**: matplotlib is a plotting library for Python that provides an object-oriented API for embedding plots into applications or generating high-quality plots in Python.

**Key Features**:

* **Plot Types**: Support for various plot types (line plots, scatter plots, histograms, etc.).
* **Customization**: High level of customization for plots (e.g., colors, labels, legends).
* **Integration**: Can be used with other libraries like pandas and numpy.

**Usage in Project**:

* Creating visualizations of ticket booking trends and model performance.
* Generating plots for exploratory data analysis.
  1. **seaborn**

**Overview**: seaborn is a Python visualization library based on matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics.

**Key Features**:

* **Statistical Plots**: Built-in functions for creating statistical plots (e.g., box plots, violin plots).
* **Integration**: Works well with pandas DataFrames.
* **Ease of Use**: Simplifies the creation of complex visualizations.

**Usage in Project**:

* Enhancing data visualizations with statistical plots.
* Visualizing correlations and distributions.

# CHAPTER 7 SYSTEM DESIGN

##### System Design

* 1. **Architecture for the "Transportation Demand Prediction" Project**

The architecture of the "Transportation Demand Prediction" project involves several key components working together to collect, process, analyze, and present ticket booking dat. Here's a high-level overview of the architecture:

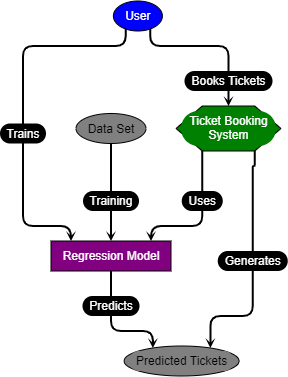


Fig: 7.1 Architecture

1. **User (Blue Oval):**
   * Represents the end-user who interacts with the ticket booking system.
2. **Books Tickets (Black Label):**
   * Indicates the user's action to book tickets through the ticket booking system.
3. **Ticket Booking System (Green Hexagon):**
   * The central component where users can book tickets.
   * This system interacts with the regression model to generate predictions.
4. **Generates (Black Label):**
   * The action taken by the ticket booking system to generate predicted tickets using the regression model.
5. **Predicted Tickets (Gray Oval):**
   * The output generated by the ticket booking system using the regression model, which predicts the likelihood of ticket booking cancellations.
6. **Uses (Black Label):**
   * Indicates that the ticket booking system uses the regression model for generating predictions.
7. **Regression Model (Purple Rectangle):**
   * Represents the machine learning model (Logistic Regression and Random Forest) used to predict ticket cancellations.
8. **Predicts (Black Label):**
   * The action taken by the regression model to predict the likelihood of ticket cancellations based on the dataset.
9. **Training (Black Label):**
   * The process of training the regression model using the provided dataset.
10. **Data Set (Gray Oval):**
    * The dataset used for training the regression model.
11. **Trains (Black Label):**
    * Indicates that the user is responsible for providing the dataset and initiating the training of the regression model.
    1. **Workflow:**
12. **Booking Tickets:**
    * The user interacts with the ticket booking system to book tickets.
    * The ticket booking system uses the regression model to predict whether the ticket will be canceled or not.
    * Based on the prediction, the system generates predicted tickets.
13. **Training the Regression Model:**
    * The user provides a dataset to train the regression model.
    * The training process involves using the dataset to train the model.
    * The trained regression model is then used by the ticket booking system to make predictions.
14. **Predicting Ticket Cancellations:**
    * The trained regression model predicts the likelihood of ticket cancellations based on the input data.
    * The ticket booking system uses these predictions to generate information
    1. **Data Science**

**Overview**: Data science involves extracting insights and knowledge from structured and unstructured data. It encompasses various techniques, including statistical analysis, machine learning, and data visualization.

**Relevance to the Project**:

* **Data Collection and Preprocessing**: The project involves gathering historical ticket booking data and preparing it for analysis.
* **Feature Engineering**: Creating and selecting relevant features to improve predictive accuracy.
* **Model Building**: Using regression algorithms to predict future ticket bookings based on historical data.
* **Evaluation and Analysis**: Assessing the performance of the predictive models and deriving actionable insights.
  1. **Machine Learning**

**Overview**: Machine learning is a subset of artificial intelligence (AI) focused on building algorithms that can learn from and make predictions or decisions based on data.

**Relevance to the Project**:

* **Regression Analysis**: The core of the project is applying various regression algorithms (e.g., Linear Regression, Polynomial Regression, Ridge Regression) to predict ticket booking demand.
* **Model Training and Evaluation**: Training models on historical data and evaluating their performance to ensure accuracy and reliability.
  1. **Predictive Analytics**

**Overview**: Predictive analytics uses statistical techniques and machine learning models to forecast future outcomes based on historical data.

**Relevance to the Project**:

* **Demand Forecasting**: The primary goal of the project is to predict future ticket bookings based on historical trends and patterns.
* **Trend Analysis**: Analyzing historical booking data to identify trends and patterns that can inform future predictions.
  1. **Web Development**

**Overview**: Web development involves creating and maintaining websites and web applications. It includes front-end (client-side) and back-end (server-side) development.

**Relevance to the Project**:

* **User Interface**: Developing a web-based interface using frameworks like Flask or Django to allow users to interact with the prediction system.
* **Deployment**: Hosting the web application on a server (e.g., Apache, Nginx) to make it accessible to users.
  1. **Software Engineering**

**Overview**: Software engineering encompasses the systematic design, development, testing, and maintenance of software applications.

**Relevance to the Project**:

* **System Design**: Designing the architecture of the ticket booking prediction system, including data flow, model integration, and user interaction.
* **Development Practices**: Implementing best practices in coding, testing, and version control (using Git and GitHub/GitLab).

1. USE CASE DIAGRAM

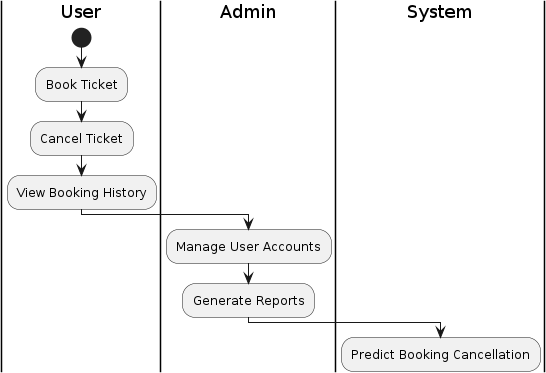


Fig: 7.2 Use Case diagram

**Explanation:**

The use case diagram for the "TRANSPORTATION DEMAND PREDICTION using Logistic Regression and Random Forest" project outlines the main functionalities and interactions between different actors (User, Admin, and System) and the system itself. Here's a breakdown of each component:

* 1. **Actors:**
     + **User:** Represents the end-users of the ticket booking system who will book tickets, cancel them, and view their booking history.
     + **Admin:** Represents the administrative users who manage user accounts and generate reports.
     + **System:** Represents the automated system components that handle predictions for booking cancellations using logistic regression and random forest algorithms.
  2. **Use Cases:**
     + **Book Ticket:** Allows users to book tickets.
     + **Cancel Ticket:** Allows users to cancel previously booked tickets.
     + **View Booking History:** Allows users to view their booking history.
     + **Manage User Accounts:** Allows admins to manage user accounts (e.g., create, update, delete accounts).
     + **Generate Reports:** Allows admins to generate various reports related to bookings.
     + **Predict Booking Cancellation:** The system uses logistic regression and random forest algorithms to predict whether a booking will be canceled.

1. CLASS DIAGRAM

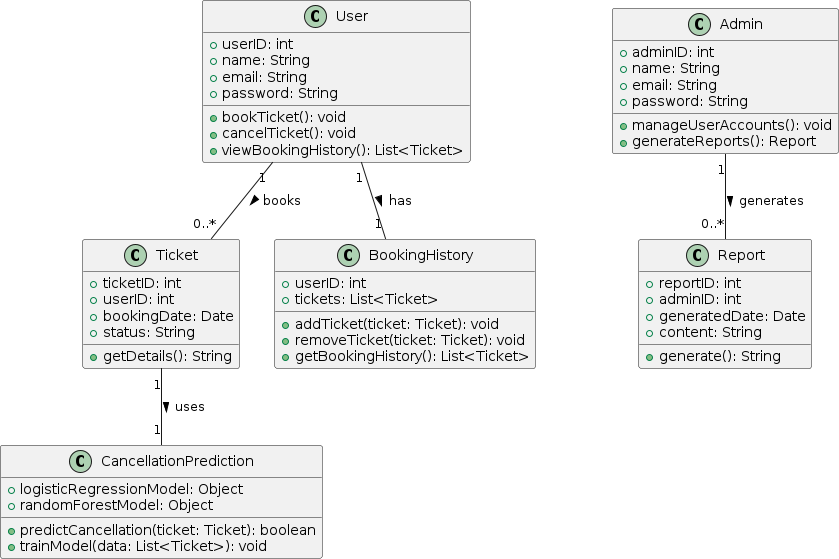


Fig: 7.3 Class Diagram

**Explanation:**

* **Classes:**
  + **User:** Represents a user of the system with attributes like userID, name, email, and password. It has methods to book and cancel tickets and view booking history.
  + **Admin:** Represents an admin with attributes like adminID, name, email, and password. It has methods to manage user accounts and generate reports.
  + **Ticket:** Represents a ticket with attributes like ticketID, userID, bookingDate, and status. It has a method to get ticket details.
  + **BookingHistory:** Represents the booking history for a user, with methods to add and remove tickets and get the booking history.
  + **CancellationPrediction:** Handles the prediction of booking cancellations using logistic regression and random forest models. It has methods to predict cancellation and train the model.
  + **Report:** Represents a report generated by an admin, with attributes like reportID, adminID, generatedDate, and content. It has a method to generate the report content.

**Associations:**

* **User books Tickets:** A User can book multiple Tickets.
* **Admin generates Reports:** An Admin can generate multiple Reports.
* **User has BookingHistory:** A User has one BookingHistory.
* **Ticket uses CancellationPrediction:** A Ticket uses the CancellationPrediction class to predict cancellation.

1. SEQUENCE DIAGRAM

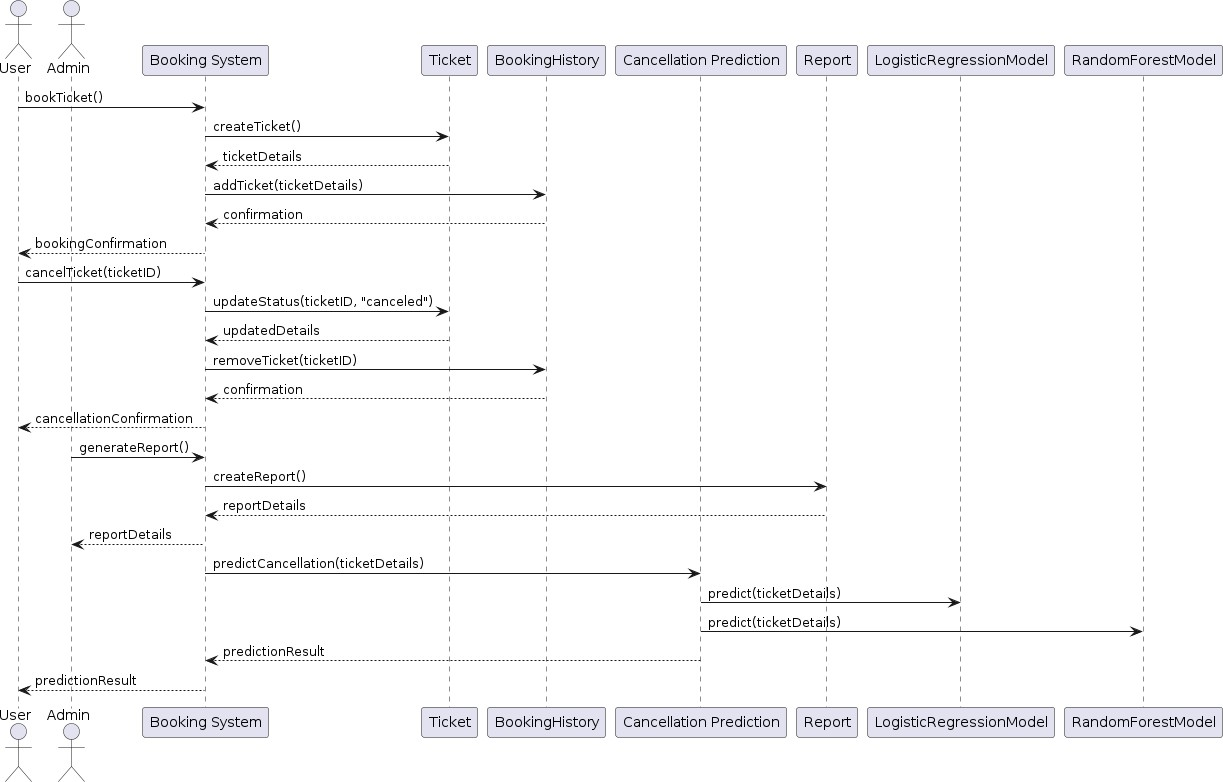


Fig: 7.4 Sequence Diagram

EXPLANATION

**Explanation:**

The sequence diagram illustrates the interactions between different components (actors and systems) during the ticket booking and cancellation process, report generation, and cancellation prediction.

* 1. **Booking a Ticket:**
     + The User initiates the bookTicket() request to the BookingSystem.
     + The BookingSystem creates a Ticket and retrieves the ticket details.
     + The BookingSystem adds the ticket to the BookingHistory.
     + The BookingHistory confirms the addition, and the BookingSystem sends a booking confirmation back to the User.
  2. **Canceling a Ticket:**
     + The User initiates the cancelTicket(ticketID) request to the BookingSystem.
     + The BookingSystem updates the ticket status to "canceled" and retrieves the updated details from Ticket.
     + The BookingSystem removes the ticket from the BookingHistory.
     + The BookingHistory confirms the removal, and the BookingSystem sends a cancellation confirmation back to the User.
  3. **Generating a Report:**
     + The Admin requests generateReport() from the BookingSystem.
     + The BookingSystem creates a Report and retrieves the report details.
     + The BookingSystem sends the report details back to the Admin.
  4. **Predicting Cancellation:**
     + The BookingSystem requests predictCancellation(ticketDetails) from the CancellationPrediction system.
     + The CancellationPrediction system uses both the LogisticRegressionModel and the RandomForestModel to predict the cancellation.
     + The prediction result is sent back to the BookingSystem, which then forwards it to the User.

1. ACTIVITY DIAGRAM

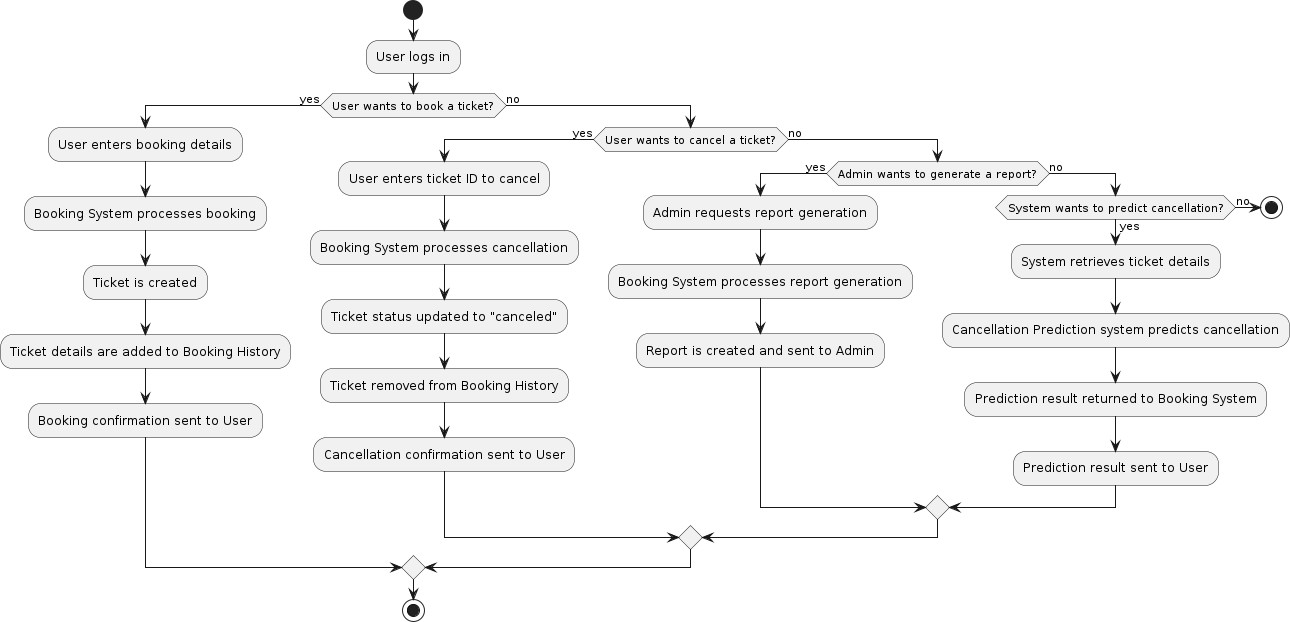


Fig: 7.5 Activity Diagram

**Explanation:**

The activity diagram illustrates the various activities and decision points in the "TRANSPORTATION DEMAND PREDICTION using Logistic Regression and Random Forest" project.

* 1. **User logs in:**
     + The starting point of the activity where the user logs into the system.
  2. **Decision Point:**
     + The system checks if the user wants to book a ticket.
       - **Yes:** The user enters booking details, the booking system processes the booking, a ticket is created, the ticket details are added to the booking history, and a booking confirmation is sent to the user.
       - **No:** The system moves to the next decision point.
  3. **Decision Point:**
     + The system checks if the user wants to cancel a ticket.
       - **Yes:** The user enters the ticket ID to cancel, the booking system processes the cancellation, the ticket status is updated to "canceled," the ticket is removed from the booking history, and a cancellation confirmation is sent to the user.
       - **No:** The system moves to the next decision point.
  4. **Decision Point:**
     + The system checks if the admin wants to generate a report.
       - **Yes:** The admin requests report generation, the booking system processes the report generation, a report is created, and the report is sent to the admin.
       - **No:** The system moves to the next decision point.
  5. **Decision Point:**
     + The system checks if it needs to predict cancellation.
       - **Yes:** The system retrieves ticket details, the cancellation prediction system predicts cancellation using logistic regression and random forest, the prediction result is returned to the booking system, and the prediction result is sent to the user.
       - **No:** The activity stops.
  6. **End:**
     + The end of the activity where the process stops.

# CHAPTER 8

**SYSTEM IMPLEMENTATION**

#### SYSTEM IMPLEMENTATION

The "Transportation Demand Prediction" project follows a structured methodology comprising several key modules. Each module plays a crucial role in developing an efficient and accurate ticket demand forecasting system. The methodology is divided into the following main modules:

1. Data Collection and Preprocessing
2. Feature Selection and Engineering
3. Model Development
4. Model Evaluation and Optimization
5. Prediction and Analysis
6. User Interface Development
   1. **Data Collection and Preprocessing**

**Objective**: Gather and prepare historical ticket booking data for analysis.

**Steps**:

* **Data Collection**: Obtain historical ticket booking data from various sources, such as airline reservation systems, railway booking systems, and entertainment ticketing platforms. The data may include details like booking date, travel date, price, customer demographics, and external factors (e.g., holidays, events).
* **Data Cleaning**: Remove inconsistencies, duplicates, and missing values from the dataset. Ensure that the data is accurate and complete.
* **Data Transformation**: Convert data into a suitable format for analysis. This may involve normalization, scaling, and encoding categorical variables.
* **Data Splitting**: Divide the dataset into training, validation, and test sets to facilitate model development and evaluation.
  1. **Feature Selection and Engineering**

**Objective**: Identify and create relevant features that influence ticket bookings.

**Steps**:

* **Feature Selection**: Determine which features (variables) are most relevant for predicting ticket demand. This may include booking date, travel date, price, day of the week, seasonality, holidays, special events, and customer demographics.
* **Feature Engineering**: Create new features that capture important patterns and relationships in the data. For example, create binary variables for holidays, calculate the number of days between booking and travel, and generate lag features to capture historical booking trends.
* **Feature Scaling**: Apply scaling techniques, such as Min-Max scaling or Standardization, to ensure that all features are on a comparable scale, which is crucial for many regression algorithms.
  1. **Model Development**

**Objective**: Develop and train regression models to predict ticket bookings.

**Steps**:

* **Linear Regression**: Implement a simple linear regression model as a baseline. This model assumes a linear relationship between the features and the target variable (ticket bookings).
* **Polynomial Regression**: Extend the linear model to capture non-linear relationships by adding polynomial terms of the features.
* **Regularized Regression Models**: Implement Ridge and Lasso regression models to handle multicollinearity and perform feature selection. These models add regularization terms to the loss function to penalize large coefficients.
* **Advanced Regression Models**: Explore other regression techniques, such as Elastic Net regression (which combines Ridge and Lasso), and tree-based models like Decision Trees and Random Forests.
  1. **Model Evaluation and Optimization**

**Objective**: Assess the performance of different models and optimize them for better accuracy.

**Steps**:

* **Evaluation Metrics**: Use metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared to evaluate model performance. These metrics provide insights into the accuracy and reliability of the predictions.
* **Cross-Validation**: Perform k-fold cross-validation to ensure the robustness and generalizability of the models. This involves dividing the training data into k subsets, training the model on k-1 subsets, and validating it on the remaining subset. Repeat this process k times and average the results.
* **Hyperparameter Tuning**: Optimize model hyperparameters using techniques like Grid Search or Random Search. Hyperparameters are parameters that are not learned from the data but set before the training process, such as the regularization strength in Ridge and Lasso regression.
* **Model Selection**: Compare the performance of different models and select the best-performing one for deployment. The selected model should have the lowest error metrics and the highest R-squared value.
  1. **Prediction and Analysis**

**Objective**: Use the optimized model to predict future ticket bookings and provide actionable insights.

**Steps**:

* **Prediction**: Apply the optimized model to the test data to generate predictions for future ticket bookings. Analyze the predicted values to identify trends and patterns.
* **Trend Analysis**: Examine the predicted booking trends to identify peak periods, seasonal variations, and the impact of external factors. This analysis can help in making informed decisions about ticket allocation and pricing.
* **Scenario Analysis**: Perform "what-if" analyses to understand how changes in features (e.g., price, special events) impact ticket demand. This can help in designing effective marketing and pricing strategies.
  1. **User Interface Development**

**Objective**: Create a user-friendly interface for stakeholders to interact with the model and view predictions.

**Steps**:

* **Interface Design**: Design a web-based interface that allows users to input new data, view predictions, and generate reports. The interface should be intuitive and easy to navigate.
* **Input Forms**: Develop forms that enable users to input relevant features, such as booking date, travel date, price, and external factors.
* **Prediction Visualization**: Implement visualization tools to display predictions and trends. Use charts and graphs to make the information easily understandable.
* **Reporting**: Create reporting features that allow users to generate and export reports based on the predictions and analyses. Reports should include key insights and recommendations for decision-making.
  1. **Expected Outcomes**

The successful implementation of this project is expected to result in:

* **Improved Prediction Accuracy**: Enhanced precision in forecasting ticket demand, reducing the likelihood of overbooking or underbooking.
* **Optimized Resource Allocation**: Better management of resources, leading to increased operational efficiency and cost savings.
* **Increased Revenue**: Implementation of dynamic pricing strategies based on demand predictions, maximizing revenue potential.
* **Enhanced Customer Satisfaction**: Improved availability of tickets during high-demand periods, leading to a better customer experience
  1. **Conclusion**

The methodology outlined above provides a comprehensive framework for developing an efficient and accurate ticket demand forecasting system using regression techniques. By following a structured approach, this project aims to enhance ticket allocation, optimize resource management, and improve overall operational efficiency across various industries. The integration of advanced regression models, combined with a user-friendly interface, will enable stakeholders to make data-driven decisions and maximize revenue potential.

* 1. **Algorithms Explanation**

The "Transportation Demand Prediction" project utilizes a variety of regression algorithms to predict ticket demand. Each algorithm has unique characteristics and is suitable for different types of data and prediction scenarios. Below is a detailed explanation of the algorithms used in this project:

* + 1. **Linear Regression**

**Overview**: Linear regression is the simplest form of regression analysis. It models the relationship between a dependent variable (target) and one or more independent variables (features) by fitting a linear equation to the observed data.

**Mathematical Representation**: y=β0+β1x1+β2x2+⋯+βnxn+ϵy = \beta\_0 +

\beta\_1x\_1 + \beta\_2x\_2 + \cdots + \beta\_nx\_n + \epsilony=β0+β1x1+β2x2+⋯+βnxn

+ϵ where:

* + - * yyy is the dependent variable.
      * x1,x2,…,xnx\_1, x\_2, \ldots, x\_nx1,x2,…,xn are the independent variables.
      * β0,β1,…,βn\beta\_0, \beta\_1, \ldots, \beta\_nβ0,β1,…,βn are the coefficients.
      * ϵ\epsilonϵ is the error term.

**Advantages**:

* + - * Simple to implement and interpret.
      * Efficient for small datasets.

**Disadvantages**:

* + - * Assumes a linear relationship between the variables.
      * Sensitive to outliers.
    1. **Polynomial Regression**

**Overview**: Polynomial regression is an extension of linear regression that models the relationship between the dependent and independent variables as an nth-degree polynomial. It can capture non-linear relationships by adding polynomial terms to the linear equation.

**Mathematical Representation**: y=β0+β1x+β2x2+⋯+βnxn+ϵy = \beta\_0 + \beta\_1x

+ \beta\_2x^2 + \cdots + \beta\_nx^n + \epsilony=β0+β1x+β2x2+⋯+βnxn+ϵ

**Advantages**:

* + - * Can model non-linear relationships.
      * More flexible than linear regression.

**Disadvantages**:

* + - * Higher risk of overfitting, especially with high-degree polynomials.
      * More complex to interpret.
    1. **Ridge Regression**

**Overview**: Ridge regression, also known as Tikhonov regularization, is a technique used to address multicollinearity in linear regression models by adding a regularization term to the loss function. This term penalizes large coefficients, reducing their impact on the model.

**Mathematical Representation**:

Minimize ∑i=1n(yi−y^i)2+λ∑j=1pβj2\text{Minimize } \sum\_{i=1}^{n} (y\_i -

\hat{y}\_i)^2 + \lambda \sum\_{j=1}^{p} \beta\_j^2Minimize ∑i=1n(yi−y^i)2+λ∑j=1p βj2 where λ\lambdaλ is the regularization parameter.

**Advantages**:

* + - * Reduces overfitting.
      * Handles multicollinearity effectively.

**Disadvantages**:

* + - * The choice of λ\lambdaλ is crucial and requires tuning.
      * Less interpretable than standard linear regression.
    1. **Lasso Regression**

**Overview**: Lasso regression (Least Absolute Shrinkage and Selection Operator) is similar to Ridge regression but uses L1 regularization. This adds an absolute value

penalty to the loss function, which can shrink some coefficients to zero, effectively performing feature selection.

**Mathematical Representation**:

Minimize ∑i=1n(yi−y^i)2+λ∑j=1p∣βj∣\text{Minimize } \sum\_{i=1}^{n} (y\_i -

\hat{y}\_i)^2 + \lambda \sum\_{j=1}^{p} |\beta\_j|Minimize ∑i=1n(yi−y^i)2+λ∑j=1p

∣βj∣

**Advantages**:

* Performs feature selection by setting some coefficients to zero.
* Helps in simplifying models.

**Disadvantages**:

* The choice of λ\lambdaλ is crucial and requires tuning.
* Can be less stable than Ridge regression.
  + 1. **Elastic Net Regression**

**Overview**: Elastic Net regression combines the penalties of both Ridge and Lasso regression. It adds both L1 and L2 regularization terms to the loss function, balancing between Ridge and Lasso.

**Mathematical Representation**:

Minimize ∑i=1n(yi−y^i)2+λ1∑j=1p∣βj∣+λ2∑j=1pβj2\text{Minimize }

\sum\_{i=1}^{n} (y\_i - \hat{y}\_i)^2 + \lambda\_1 \sum\_{j=1}^{p} |\beta\_j| +

\lambda\_2 \sum\_{j=1}^{p} \beta\_j^2Minimize ∑i=1n(yi−y^i)2+λ1∑j=1p∣βj∣+λ2

∑j=1pβj2

**Advantages**:

* Combines the benefits of Ridge and Lasso regression.
* Useful when dealing with highly correlated features.

**Disadvantages**:

* Requires tuning of two hyperparameters (λ1\lambda\_1λ1 and λ2\lambda\_2λ2).
* More complex than individual Ridge or Lasso regression.
  + 1. **Decision Trees**

**Overview**: Decision trees are non-linear models that split the data into subsets based on feature values, making decisions at each node to minimize prediction error. They can capture complex relationships between variables.

**Mathematical Representation**: A decision tree model recursively splits the feature space into regions. Each split is chosen to minimize a loss function (e.g., mean squared error for regression).

**Advantages**:

* + - * Can model complex, non-linear relationships.
      * Easy to interpret and visualize.

**Disadvantages**:

* + - * Prone to overfitting, especially with deep trees.
      * Can be unstable with small changes in data.
    1. **Random Forests**

**Overview**: Random forests are ensemble models that combine multiple decision trees to improve prediction accuracy and reduce overfitting. Each tree is built on a random subset of the data and features.

**Mathematical Representation**: The final prediction is the average (for regression) of the predictions from all individual trees.

**Advantages**:

* + - * Reduces overfitting compared to individual decision trees.
      * Handles large datasets and many features well.

**Disadvantages**:

* + - * More complex and computationally intensive.
      * Less interpretable than a single decision tree.

**Conclusion**

Each algorithm has its strengths and weaknesses, and the choice of algorithm depends on the specific characteristics of the ticket booking data and the requirements of the prediction task. Linear and polynomial regressions provide simplicity and ease of interpretation

* 1. **TESTING**

1. **Types of Testing**

For the "Transportation Demand Prediction" project, the following types of testing are particularly relevant:

* 1. **Unit Testing Focus**:
     + **Objective**: Test individual components or functions to ensure they work correctly in isolation.
     + **Suitable For**: Validating functions within the data preprocessing, prediction model, and web interface modules.
  2. **Integration Testing Focus**:
     + **Objective**: Test the interactions between different modules or components to ensure they work together as expected.
     + **Suitable For**: Ensuring that data flows correctly between the data collection module, predictive modeling, and web interface.
  3. **Functional Testing Focus**:
     + **Objective**: Verify that the system meets the specified functional requirements.
     + **Suitable For**: Testing prediction accuracy, data input forms, and result display functionalities.
  4. **Performance Testing Focus**:
     + **Objective**: Evaluate the system’s responsiveness and stability under load.
     + **Suitable For**: Testing system performance during peak usage times and validating response times for prediction requests.
  5. **Usability Testing Focus**:
     + **Objective**: Assess how user-friendly and intuitive the web interface is.
     + **Suitable For**: Ensuring ease of navigation, form usability, and overall user experience.
  6. **Security Testing Focus**:
     + **Objective**: Identify and address security vulnerabilities.
     + **Suitable For**: Testing user authentication, data protection, and authorization mechanisms.
  7. **Regression Testing Focus**:
     + **Objective**: Ensure that new changes or additions do not negatively impact existing functionality.
     + **Suitable For**: Testing the overall system after updates or modifications to any module.
  8. **Acceptance Testing Focus**:
     + **Objective**: Verify that the system meets the business requirements and is ready for deployment.
     + **Suitable For**: Validating end-to-end functionality and ensuring the system meets user expectations.

1. **Test Cases for the Project**

Here’s a table format for some of the test cases suitable for the project:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Module** | **Test Description** | **Expected Result** | **Status** |
| TC001 | Data Collection | Verify data ingestion from CSV file | Data is correctly ingested and stored in the database | Not Run |
| TC002 | Data Preprocessing | Test handling of missing values | Missing values are filled or removed appropriately | Not Run |
| TC003 | Prediction Model | Validate model  training with historical data | Model trains without errors and performs as expected | Not Run |
| TC004 | Prediction Model | Check model prediction accuracy | Predictions match expected values with acceptable error | Not Run |
| TC005 | Web Interface | Test data input form submission | Form data is correctly processed and results are displayed | Not Run |
| TC006 | Web Interface | Verify result display and visualizations | Results and visualizations are displayed correctly | Not Run |
| TC007 | Performance | Measure system | Response time is within the | Not |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Module** | **Test Description** | **Expected Result** | **Status** |
|  |  | response time for predictions | acceptable limit (e.g., 5 seconds) | Run |
| TC008 | Security | Test user  authentication and authorization | Access is restricted based on user roles and credentials | Not Run |
| TC009 | Usability | Assess ease of navigation and form usability | Interface is intuitive and easy to navigate | Not Run |
| TC010 | Integration | Check data flow between modules | Data flows correctly between data collection, preprocessing, and prediction modules | Not Run |

# CHAPTER 9 RESULTS

#### RESULTS:

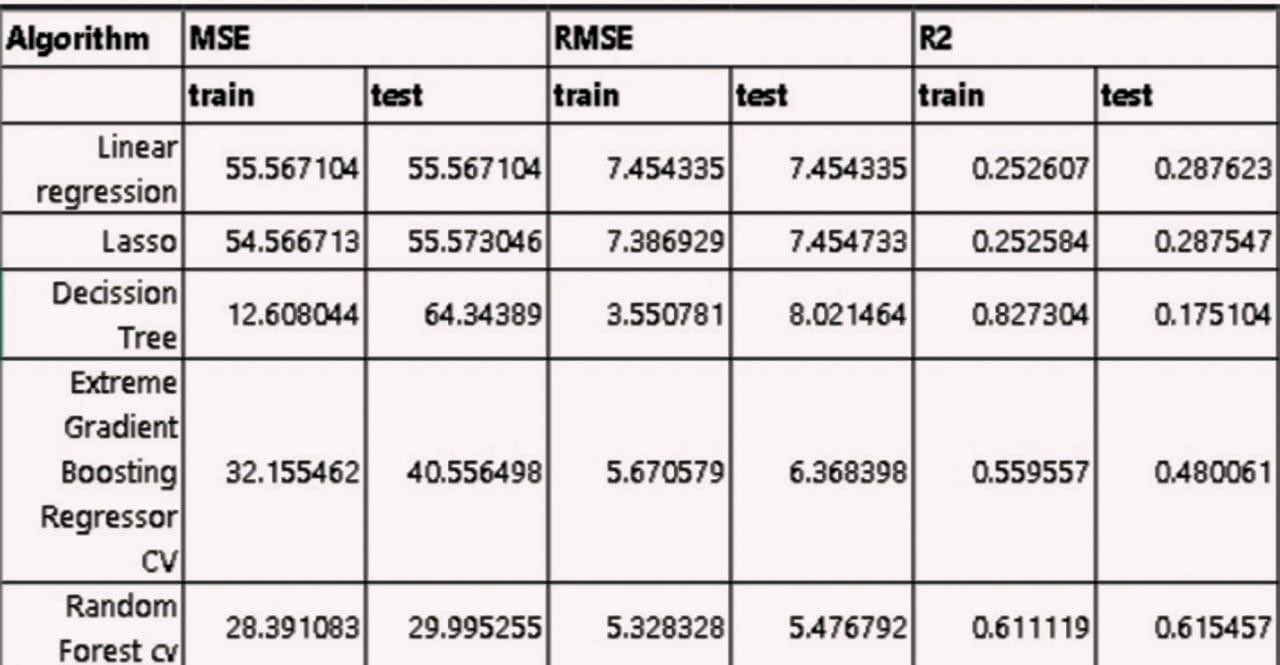


Fig: 9.1 Results

This table presents the performance of various regression models using three key metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R- squared (R2). Lower MSE and RMSE values indicate better model fit, as they represent the average squared and root-squared differences between predicted and actual values, respectively. R2, on the other hand, measures the proportion of variance in the dependent variable that is predictable from the independent variables, with higher values (closer to 1) indicating a better fit.

Random Forest CV is the most suitable model for transportation demand prediction due to its ability to handle complex, non-linear relationships often found in transportation data. It effectively reduces overfitting by averaging multiple decision trees, enhancing its predictive accuracy and robustness. The consistently high R2 value on the test data indicates that it can accurately capture the variability in transportation demand, making it reliable for forecasting. Additionally, its ability to handle a mix of categorical and numerical data, which is common in transportation datasets, further contributes to its superior performance in this context

The model has predicted well at training. I found that for training data there is a R2 score of 61.11% which means the model has predicted well on the datapoints. Which means low bias. Also the model has a R2 score of 61.54%. The variance in model is also low. This is the best performing model I have found.

Therefore, based on these metrics, the **Random Forest CV** model would be the preferred choice for this regression task due to its superior performance in terms of minimizing error and maximizing the explained variance on the test data.

* 1. **USER INTERFACE:**



Fig: 9.2 User Interface

1. **Title Section**
   * The title "Ticket Number Prediction App" is displayed at the top in bold, making it clear to users what the application is about.
2. **Input Fields**

The UI contains multiple input fields, each designed to collect specific data for ticket prediction:

* + **Select Date**: Allows users to choose a date for prediction.
  + **Max Capacity**: Defines the total number of tickets available.
  + **Travel From Distance (in km)**: Captures the distance from the origin location.
  + **Time Period (24-hour format)**: Represents the hour of travel, affecting demand.
  + **Waiting Time (in minutes)**: The time a user waits before departure.
  + **Speed (in km/h)**: The travel speed, which may influence demand.

Each input field has:

* + A numeric value display (initially set to 0).
  + Increment (+) and decrement (-) buttons to adjust values easily.

1. **Additional Information**

Below the input fields, the app displays:

* + **Day, Weekday, and Day of Year**: Extracted from the selected date, possibly used for demand calculations.

1. **Prediction Button**

A **"**Predict Number of Tickets**"** button (not visible in this specific screenshot but seen in the previous one) triggers the prediction based on the provided input values.

1. **Predicted Output (Not Visible)**
   * In the previous image, the app displayed the predicted number of tickets after calculation.
   * In this image, no prediction result is shown, likely because the input values are all zero.

**User Experience and Design**

* + **Simple and clean layout**: Users can easily navigate and input values.
  + **Interactive elements**: Buttons for easy adjustments.
  + **Automated calculations**: The app extracts date-related information and provides a prediction.
  1. **OUTPUT 1:**

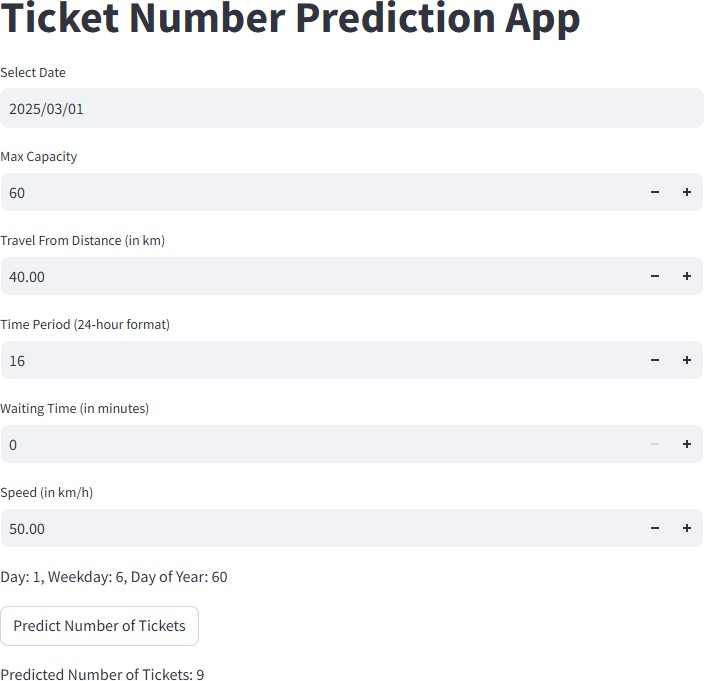


Fig: 9.3 Output 1

The Ticket Number Prediction App in the image calculates the expected number of tickets based on several input parameters. The predicted output is 9 tickets, which is displayed at the bottom.

**Key Inputs and Their Role in the Prediction:**

1. **Selected Date:**
   * The date chosen is March 1, 2025.
   * The app extracts Day (1), Weekday (6 - Saturday), and Day of Year

(60) from the date, likely using them to factor in demand trends.

1. **Max Capacity: 60**
   * The total number of available tickets is **60**, setting an upper limit on sales.
2. **Travel Distance: 40 km**
   * The model may consider that longer travel distances impact ticket demand.
   * A moderate distance (40 km) could indicate a regional or short intercity journey.
3. **Time Period: 16 (4:00 PM)**
   * Demand can vary based on the time of travel.
   * 4:00 PM might fall into an afternoon peak period, depending on location trends.
4. **Waiting Time: 0 minutes**
   * No waiting time might indicate immediate boarding, potentially making the trip more attractive.
5. **Speed: 50 km/h**
   * Travel speed can affect journey duration, influencing passenger preferences.
   * A speed of 50 km/h suggests a moderate pace, possibly for a bus or local train service.
   1. **OUTPUT 2:**

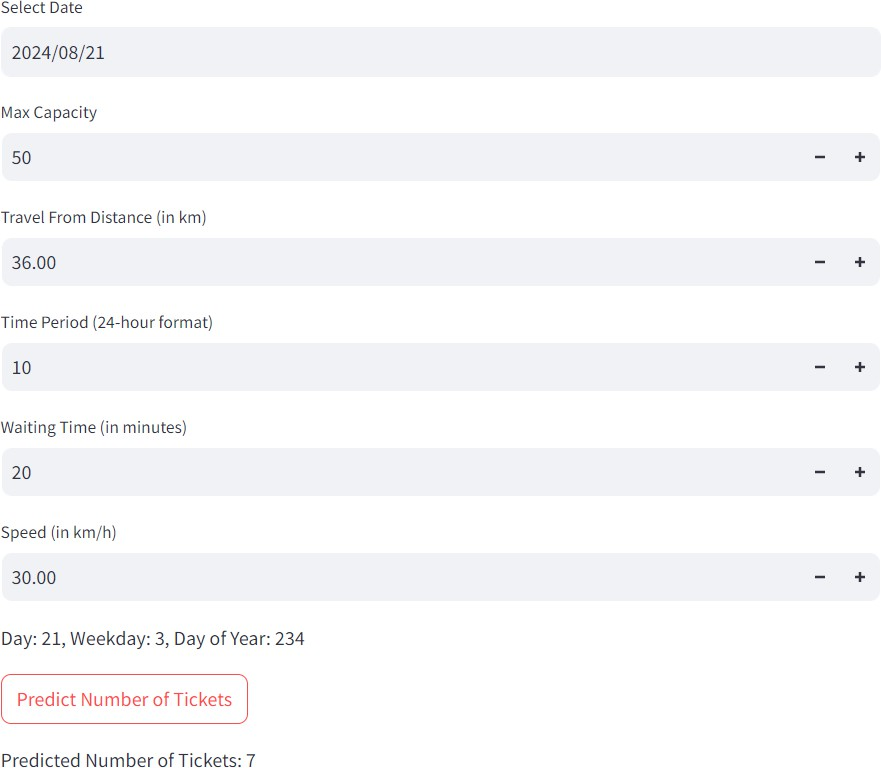


Fig: 9.4 Output 2

The Ticket Number Prediction App in the image calculates the expected number of tickets based on several input parameters. The predicted output is 9 tickets, which is displayed at the bottom.

**Key Inputs and Their Influence on the Prediction:**

1. **Selected Date: August 21, 2024**
   * The app determines that this corresponds to **Day 234 of the year** and

**Weekday 3 (Wednesday)**.

* + Midweek demand may be lower than weekends, potentially affecting ticket sales.

1. **Max Capacity: 50**
   * The total number of available tickets is **50**, meaning the maximum possible sales.
2. **Travel Distance: 36 km**
   * A moderate distance, possibly influencing ticket demand.
   * Longer distances might reduce demand if other travel alternatives exist.
3. **Time Period: 10 (10:00 AM)**
   * Travel demand varies by time of day.
   * **Mid-morning (10 AM)** may have lower demand compared to peak commute times (early morning or evening).
4. **Waiting Time: 20 minutes**
   * Longer waiting times can negatively impact demand as passengers prefer shorter wait times.
   * A **20-minute wait** might discourage some travelers, reducing expected ticket sales.
5. **Speed: 30 km/h**
   * A relatively slow speed, possibly indicating a local or urban route with traffic congestion.
   * Slower travel speeds may reduce passenger preference, leading to lower demand.

# CHAPTER 10 CONCLUSION

##### Conclusion

The "Transportation Demand Prediction" project leverages predictive modeling to forecast ticket bookings based on historical data. By implementing various regression algorithms and integrating them into a user-friendly web interface, the project aims to provide accurate predictions and valuable insights for booking management.

The project involves comprehensive testing to ensure reliability, performance, and usability, including unit, integration, functional, performance, usability, security, regression, and acceptance testing. Future improvements could include enhanced modeling techniques, real-time data processing, mobile support, and advanced analytics.

With a clear focus on robust system design, effective data handling, and user-friendly interactions, the project aims to deliver a reliable and insightful ticket booking prediction tool that meets the needs of its users and stakeholders.

* 1. **Future Scope**
     1. **Enhanced Models**:
        + **Incorporate Advanced Algorithms**: Explore more advanced machine learning algorithms (e.g., neural networks) to improve prediction accuracy.
        + **Real-Time Data Processing**: Implement real-time data processing and prediction features.
     2. **Broader Data Sources**:
        + **Integration with External APIs**: Integrate with external data sources or APIs for additional booking-related data.
     3. **User Personalization**:
        + **Customized Predictions**: Develop features for personalized predictions based on user history and preferences.
     4. **Mobile Application**:
        + **Mobile Support**: Create a mobile app version of the system for easier access and usability on mobile devices.
     5. **Advanced Analytics**:
        + **Detailed Analytics**: Implement more advanced analytics features, such as predictive analytics for pricing and inventory management.
     6. **Internationalization**:
        + **Multi-Language Support**: Extend the system to support multiple languages and currencies for global usage.

# CHAPTER 11 REFERENCE

1. **References**
   1. Pattern Recognition and Machine Learning" by Christopher M. Bishop.
   2. Introduction to Machine Learning with Python" by Andreas C. Müller and Sarah Guido.
   3. “Optimizing bus ticket sales operation” by Asser Letsatsi Tau, Emmanuel Innocents Edoun and Anup Pradhan
   4. "Transportation Demand Forecasting: Concepts and Methods" by Michael D. Meyer and Eric J. Miller (2001)
   5. "Demand Forecasting for Transportation" by the Transportation Research Board (2014)
   6. "Travel Demand Modeling : A Primer" by the Federal Highway Administration (2019)
   7. "Transportation Demand Management: A Guide for Practitioners" by the Victoria Transport Policy Institute (2020)
   8. "Forecasting Transportation Demand: A Review of Methods and Applications" by the International Journal of Transportation Science and Technology (2020)
   9. “ARIMA (Auto Regressive Integrated Moving Average)” have been commonly employed to analyse and predict future bookings based on historical data. Considers by Box and Jenkins (1976)
   10. Tarek S. Sayed, et al."Forecasting Urban Travel Demand: A Hybrid Model Approach"

# CHAPTER 12 APPENDIX

###### APPENDIX

# Import necessary libraries import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

# Step 1: Load the dataset

# Assuming the dataset is a CSV file with columns: 'booking\_date', 'travel\_date', 'price', 'holidays', 'events', 'bookings'

data = pd.read\_csv('ticket\_bookings.csv')

# Step 2: Data preprocessing

# Convert 'booking\_date' and 'travel\_date' to datetime data['booking\_date'] = pd.to\_datetime(data['booking\_date']) data['travel\_date'] = pd.to\_datetime(data['travel\_date'])

# Calculate the number of days between booking and travel data['days\_between'] = (data['travel\_date'] - data['booking\_date']).dt.days

# Encode categorical variables (holidays and events) as binary data['holidays'] = data['holidays'].astype(int)

data['events'] = data['events'].astype(int)

# Drop the original date columns

data.drop(['booking\_date', 'travel\_date'], axis=1, inplace=True)

# Step 3: Define features (X) and target variable (y)

X = data.drop('bookings', axis=1) # Features: all columns except 'bookings' y = data['bookings'] # Target variable: 'bookings'

# Step 4: Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Step 5: Train the Linear Regression model model = LinearRegression() model.fit(X\_train, y\_train)

# Step 6: Make predictions on the test set y\_pred = model.predict(X\_test)

# Step 7: Evaluate the model

mae = mean\_absolute\_error(y\_test, y\_pred) mse = mean\_squared\_error(y\_test, y\_pred) r2 = r2\_score(y\_test, y\_pred)

# Step 8: Print evaluation metrics print(f'Mean Absolute Error (MAE): {mae}') print(f'Mean Squared Error (MSE): {mse}') print(f'R-squared (R2): {r2}')

###### RESEARCH PAPER

**Leveraging Random Forest for Accurate Transportation Demand Forecasting**

Tangudubilli Santhoshi Lakshmi1, Karimisetty Sujatha1 , Ella Kumar Swamy*,*

Kona Anand, Dosuru Sravanthi, Villuri Sasi Vardhan,

*Dadi Institute of Engineering & Technology, Anakapalli, India*.

[*lakshmisanthoshi1@gmail.com,*](mailto:lakshmisanthoshi1@gmail.com)[*sujathakota29@gmail.com,*](mailto:sujathakota29@gmail.com)[*kumarswamyella@gmail.com*](mailto:kumarswamyella@gmail.com) *,* [*saianandkumar2002@gmail.com,*](http://saianandkumar2002@gmail.com/)[*dosurusravanthi@gmail.com,*](mailto:dosurusravanthi@gmail.com)[*sasivardhan2003@gmail.com*](mailto:sasivardhan2003@gmail.com)

**Abstract:**

Transportation Demand Forecasting is a critical component for effective urban planning and infrastructure management. Accurate forecasting of transportation needs allows for optimized resource allocation, improved service delivery, and enhanced sustainability .Existing model of our project DA4PT (Data Analytics For Public Transport), for finding the variables that impact travellers in booking and obtaining transport tickets. DA4PT focuses only on analysing transportation data to inform planning decisions, uses advanced techniques, such as machine learning and statistical modelling, to predict future transportation demand. Random Forest Regression model is used to examine correlations between factors and ticket sales using historical booking data and offers a reliable foundation for forecasting ticket sales by Mean Absolute Error (MAE) of, Mean Squared Error (MSE), R-squared value of, RMSPE and by various statistical metrics. We can deploy the model with Random Forest Regressor with the parameters with 3-fold cross validation. I found that for training data there is a R2 score of 63.46% which means the model has predicted well on the datapoints. Which means low bias. Also, the model has a R2 score of 63.93%. The variance in model is also low. The ticket booking platform can enhance revenue by utilizing these insights to optimize methods and generate additional revenue streams by integration these data into their business plan.

**Keywords**: Transportation, Travel Time Analysis, Random Forest Regression, Revenue Sales Prediction

## Introduction

The project "Leveraging Random Forest for Accurate Transportation Demand Forecasting" addresses the growing need for efficient and digital ticket management systems used across different industries, such as airlines, railways, and entertainment. Ticket booking trends are influenced by so many factors, including seasonal changes, special events, pricing strategies, and customer needs. Traditional methods of managing ticket allocation often fall short in accurately predicting demand, leading to either overbooking or underutilization of resources. This project leverages the power of regression analysis to develop a predictive model that optimizes ticket booking and enhances overall operational efficiency.

* 1. **Background**

Transport booking platforms have advanced greatly over the years, transitioning from manual processes to advanced digital platforms. Despite these advancements, many platforms still rely on basic statistical methods and outdated approaches that lack the precision required for accurate demand

forecasting. The advent of big data and machine learning presents an opportunity to revolutionize ticket booking by applying advanced regression techniques to predict future trends based on historical data. The " Transportation Demand Forecasting " project aims to develop a predictive system that forecasts future ticket bookings based on historical data. The system leverages regression analysis to predict and providing demand trends that give us valuable insights for optimizing booking strategies. By analysing past booking patterns and external factors, the project intends to enhance booking accuracy and resource utilization, ultimately resulting in better customer satisfaction and operational efficiency. The project includes several key components, such as data collection and pre-processing , predictive modelling, and a user-friendly web interface. The integration of these components aims to create a cohesive system that delivers reliable and actionable predictions to users.

* 1. **Motivation**

The inspiration for this project comes from the challenges faced by businesses and organizations in accurately forecasting ticket bookings. In the competitive travel and entertainment industries, precise demand predictions are crucial for optimizing inventory, pricing strategies, and resource allocation. Traditional methods often struggle with accuracy and efficiency, leading to overbooking or under booking issues… By implementing regression techniques, the project aims to tackle these challenges and offer a data-driven approach to booking predictions. This not only enhances operational efficiency but also enhances the overall customer experience by reducing discrepancies between actual and predicted bookings*.*

* 1. **Problem Definition**

The Initial problem addressed by this project is the lack of accurate and reliable forecasting for ticket bookings. Current systems may rely on simplistic or outdated methods that fail to account for complex patterns and trends in booking data. This can lead to inefficiencies in inventory management, revenue optimization, and customer satisfaction. Specially, the problem include current methods may not fully capture the complexities of booking trends, leading to unreliable forecasts. Poor predictions can result in either overbooking or under booking affecting operational efficiency and customer satisfaction. Traditional systems may not fully take advantage of historical data or external factors that influence booking patterns.

* 1. **Objective of the Project**

The primary objectives of the project to create a regression-based model that accurately forecasts future ticket bookings based on historical data. Enhance the accuracy of booking predictions to improve resource allocation and reduce booking inconsistencies. Design a user-friendly web interface that lets users easily input data, view predictions, and access insights. Offer tools for trend analysis and visualization to help users understand booking patterns and make informed decisions.

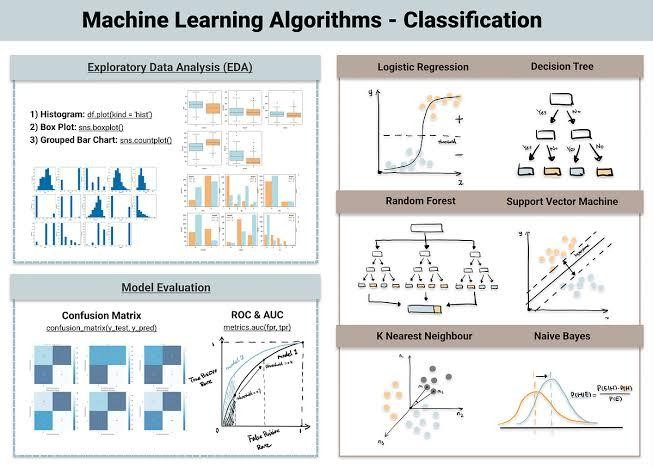
## .Literature Survey

The literature review for the " Transportation Demand Forecasting " project examines previous research and methodologies applied in the field of demand forecasting and ticket management. This review covers traditional approaches, advanced machine learning techniques, and specific applications within industries such as airlines, railways, and entertainment.

* 1. **Traditional Approaches:**

Time series analysis is commonly used for forecasting ticket demand. Techniques like ARIMA (Auto Regressive Integrated Moving Average) have been commonly employed to analyse and predict future

bookings based on historical data. Considers by Box and Jenkins (1976) highlighted the viability of ARIMA in capturing regular designs and patterns in time arrangement information. Simple linear regression and moving average methods have also been used for ticket demand forecasting. While these methods are simple to implement, they often fail to capture complex patterns and interactions between multiple influencing factors.



**Fig1.** Machine Learning Algorithms

* 1. **Machine Learning Techniques:**

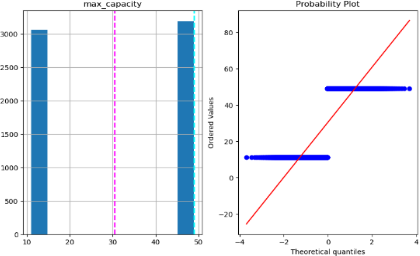
The application of various regression models has gained popularity in recent years. Linear regression, polynomial regression, and ridge regression have shown promise in predicting ticket demand by considering multiple features simultaneously. Studies by Draper and Smith (1998) demonstrated the use of linear regression for predicting airline ticket sales, showing that it can deliver reasonable accuracy when the relationship between variables is linear. Research by Aiken and West (1991) indicated that polynomial regression could capture more complex relationships, although it requires careful tuning to avoid overfitting. Techniques like Ridge and Lasso regression, introduced by Hoerl and Kennard (1970) and Tibshirani (1996) respectively, address multicollinearity and feature selection, providing more robust models for prediction.

## Methodology

The "Ticket Booking Using Regression" project follows a structured methodology comprising several key modules. Each module plays a vital role in developing an efficient and accurate ticket demand forecasting system. [4] "Transportation Demand Forecasting: Concepts and Methods" by Michael D. Meyer and Eric J. Miller (2001).

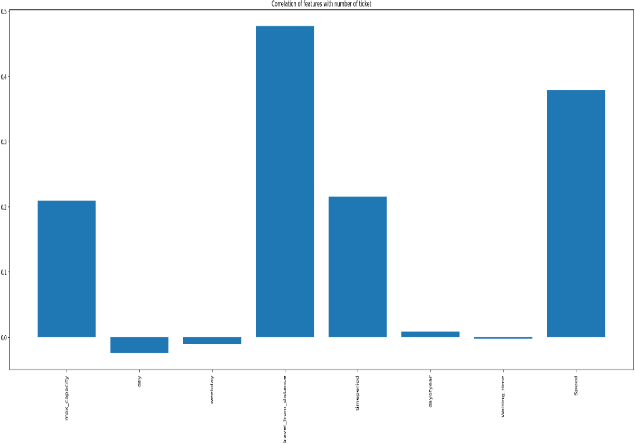
The methodology is divided into the following main modules:

* + - 1. **Data Collection and Preprocessing:**



**Fig2.** Data Transformation

* + - 1. **Objective:** Gather and prepare historical ticket booking data for analysis.Obtain historical ticket booking data from multiple sources, including airline reservation systems, railway booking systems, and entertainment ticketing platforms. The data may consist of information such as booking date, travel date, price, customer demographics, and external factors (e.g., holidays, events). Remove inconsistencies, duplicates, and missing values from the dataset. Guarantee that the information is precise and complete. Change over information into a appropriate organize for examination. This may involve normalization, scaling, and encoding categorical variables. Partition the dataset into preparing, approval, and test sets to encourage demonstrate improvement and assessment.
      2. **Feature Selection and Engineering**



**Fig3.** Feature Selection

Determine which features (variables) are the most important for predicting ticket demand. This may include booking date, travel date, price, day of the week, season, holidays, and special events, and customer demographics. Create new features that capture key patterns and

relationships within the data. For example, create binary variables for holidays, calculate the number of days between booking and travel, and generate lag features to capture historical booking trends. Apply scaling techniques like Min-Max scaling or standardization to ensure all features are on a comparable scale, which is essential for the performance of many regression algorithms.

* + - 1. **Model Development**

Objective: Develop and train regression models to predict ticket bookings. Implement a simple linear regression model as a baseline. This model assumes a linear relationship between the features and the target variable (ticket bookings). Expand the straight demonstrate to capture non-linear connections by including polynomial terms of the highlights. Implement Ridge and Lasso regression models to handle multi- collinearity and perform feature selection. These models add regularization terms to the loss function to penalize large coefficients. Explore other regression techniques, such as Elastic Net

regression (which combines Ridge and Lasso), and tree-based models like Decision Trees and Random Forests.

* + - 1. **Model Evaluation and Optimization**

Objective: Evaluate how well various models perform and adjust them for increased accuracy. Utilize measurements such as Cruel Supreme Blunder (MAE), Cruel Squared Mistake (MSE), and R-squared to assess show execution. To make sure the models are resilient and generalizable, run k-fold cross- validation. Optimize show hyperparameters utilizing methods like Lattice Look or Irregular Look. Hyperparameters, like the regularization strength in Ridge and Lasso regression, are settings made prior to training and are not learned from the data. Choose the model that performs the best for deployment by comparing its performance with other models. The selected model should have the lowest error metrics and the highest R-squared value.

* + - 1. **Prediction and Analysis**

Objective: Use the optimized model to predict future ticket bookings and provide actionable insights. Apply the optimized model to the test data to generate predictions for future ticket bookings.

## Data Set



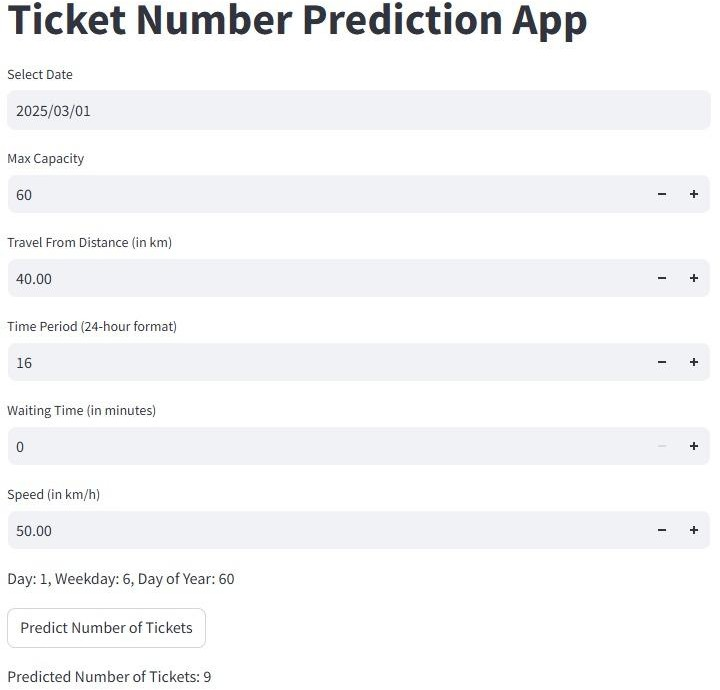
* + 1. Results



**Table1**. Train And Test values of Models

**Description:** We can deploy the model with Random Forest Regressor with the parameters

{'bootstrap': True,'max\_depth': None,'max\_features': 'sqrt','min\_samples\_leaf': 2,'min\_samples\_split': 12,'n\_estimators': 400} with Three fold cross validation. The model has predicted well at training. I found that for training data there is a R2 score of 63.46% which means the model has predicted well on the datapoints. Which means low bias. Also the model has a R2 score of 63.93%. The variance in model is also low. This is the best performing model I have found.The following outcomes are anticipated if this project is implemented successfully Enhanced precision in forecasting ticket demand, reducing the likelihood of overbooking or under booking. Better management of resources, leading to increased operational efficiency and cost savings. Implementation of dynamic pricing strategies based on demand predictions, maximizing revenue potential. Improved availability of tickets during high- demand periods, leading to a better customer experience.



**Fig7.** Ticket Number of Prediction App that acurrately determines the number of tickets as nine(9)

## Conclusion

The " Transportation Demand Forecasting " project leverages predictive modeling to forecast ticket bookings based on historical data. By implementing various regression algorithms and integrating them into a user-friendly web interface, the project aims to provide accurate predictions and valuable insights

for booking management. The project includes extensive testing, such as unit, integration, functional, performance, usability, security, regression, and acceptance testing, to guarantee dependability, performance, and usability Improved modeling methods, real-time data processing, mobile support, and sophisticated analytics are possible future additions. With a clear focus on robust system design, effective data handling, and user-friendly interactions, the project aims to deliver a reliable and insightful ticket booking prediction tool that meets the needs of its users and stakeholders.

## Future Scope

Explore more advanced machine learning algorithms (e.g., neural networks) to improve prediction accuracy. Use real-time data processing and make accurate predictions.. Connect to outside data sources or APIs for additional booking-related data. Develop features for personalized predictions based on user history and preferences. Create a mobile app version of the system for easier access and usability on mobile devices. Implement more sophisticated analytics capabilities, like predictive analytics for pricing and inventory management. Expand the system's language and currency support for widespread use.

## 1 References:

1. Pattern Recognition and Machine Learning" by Christopher M. Bishop.
2. Introduction to Machine Learning with Python" by Andreas C. Müller and Sarah Guido.
3. “Optimizing bus ticket sales operation” by Asser Letsatsi Tau, Emmanuel Innocents Edoun and Anup Pradhan
4. "Transportation Demand Forecasting: Concepts and Methods" by Michael D. Meyer and Eric J. Miller (2001)
5. "Demand Forecasting for Transportation" by the Transportation Research Board (2014)
6. "Travel Demand Modeling : A Primer" by the Federal Highway Administration (2019)
7. "Transportation Demand Management: A Guide for Practitioners" by the Victoria Transport Policy Institute (2020)
8. "Forecasting Transportation Demand: A Review of Methods and Applications" by the International Journal of Transportation Science and Technology (2020)
9. “ARIMA (Auto Regressive Integrated Moving Average)” have been commonly employed to analyse and predict future bookings based on historical data. Considers by Box and Jenkins (1976)

**CERTIFICATES OF CONFERENCE**

