BIKE SHARING DEMAND PREDICTION

Detailed Project Report

July 14, 2024

Rishabh Barman

Abstract

With the increasing availability of large, transport-related datasets, detailed data-driven mobility analysis is now possible. Publicly available databases collect trip details, including origins, destinations, and travel times, enabling precise demand forecasting. The primary challenge is ensuring the availability and accessibility of rental bikes at the right time to minimize waiting times and maintain a stable supply. This involves anticipating demand fluctuations due to factors like time of day, weather conditions, and special events. Accurate predictions allow operators to redistribute bikes proactively, reducing the risk of shortages and surpluses at different stations.

The bike share prediction system project aims to develop an accurate and adaptable machine learning model to forecast bike share demand in urban areas. By leveraging historical bike share data, weather conditions, and urban traffic accident records, the project integrates multiple data sources to enhance prediction accuracy. The model employs various regression algorithms, evaluated using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared (R2), and Root Mean Squared Error (RMSE). After extensive data processing, feature engineering, and exploratory data analysis, the best-performing model was deployed through a user-friendly web application using Flask and Streamlit. This application provides real-time demand predictions, assisting city planners and bike share operators in optimizing bike distribution and enhancing user satisfaction. The project addresses challenges such as data quality, feature selection, and real-time adaptation, proposing solutions like automated data cleaning, advanced feature engineering, and adaptive learning mechanisms. Future work involves further improving the model's adaptability, incorporating additional data sources, and integrating user feedback for continuous enhancement. This comprehensive approach ensures the system's reliability and effectiveness in managing urban bike share programs

Table of Contents

1. Introduction

The bike share prediction system aims to forecast the demand for bike shares in urban areas. Accurate predictions help optimize the distribution and availability of bikes, reduce waiting times, and enhance user satisfaction. With the rise in popularity of bike-sharing programs in cities worldwide, efficient management of these systems has become crucial. This project leverages historical bike share data, weather conditions, and urban traffic accident data to build a robust prediction model. By incorporating multiple data sources and advanced machine learning techniques, the project seeks to provide city planners and bike share operators with actionable insights to improve operational efficiency and user experience

1.1 Project Objectives

- Develop a machine learning model to predict bike share demand.
- Ensure the model can adapt in real-time to changing conditions.
- Integrate the prediction model into a web application for user-friendly access.
- Evaluate the model's performance using multiple regression algorithms and appropriate metrics.

1.2 Scope

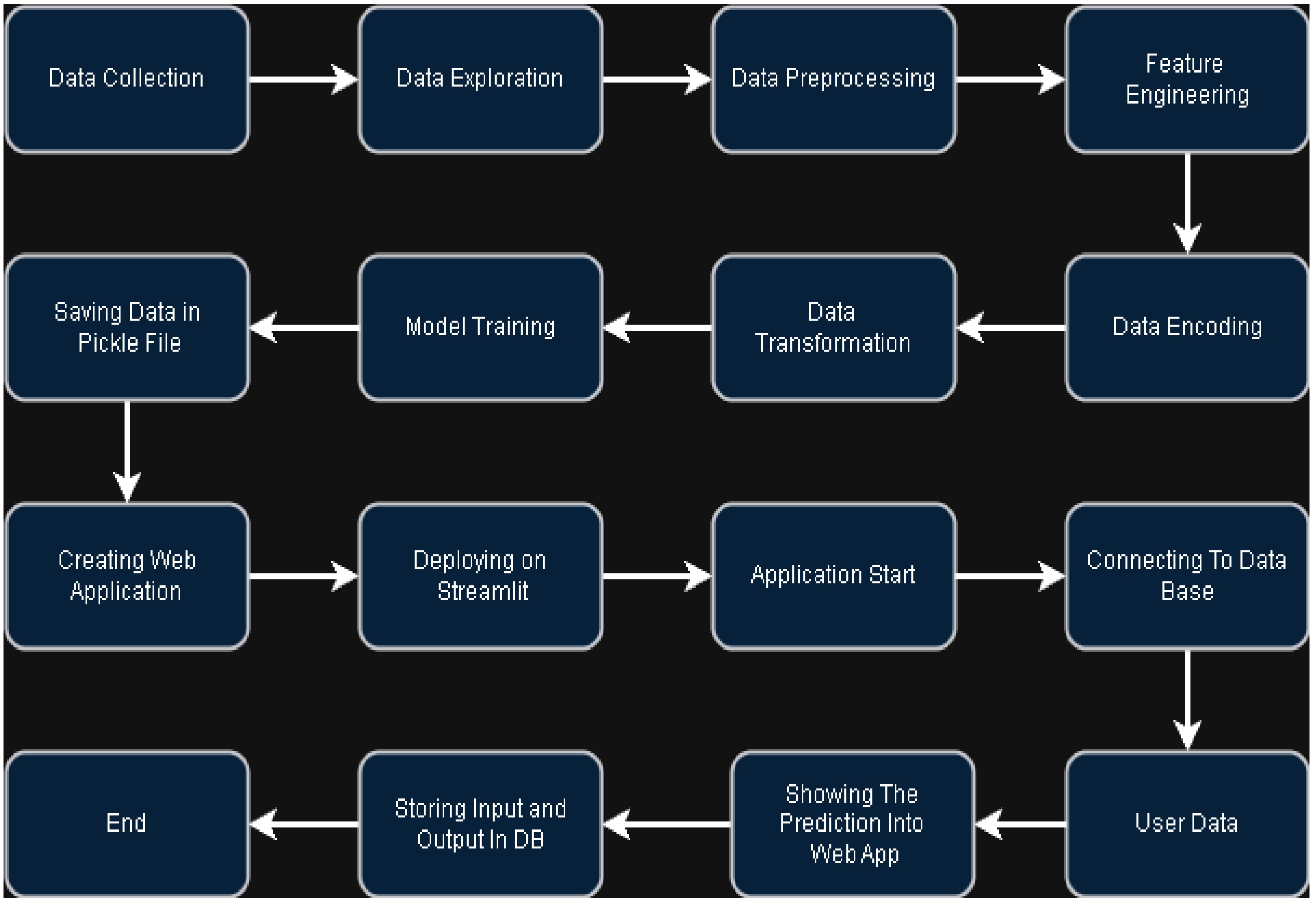
The scope of this project encompasses the development, evaluation, and deployment of a machine learning model to predict bike share demand. It includes:

- Data Integration and Processing
- Feature Engineering
- Model Training and Evaluation
- Real-time Adaptation
- Web Application Development.
- Optimization and Feedback

2. Architecture Diagram

The diagram outlines the workflow for a bike share prediction system, illustrating each step from data collection to deployment and user interaction. It consists of several key components, including data gathering, data preprocessing, model building, user interaction, and deployment

Figure 1. Architecture Diagram



3. Detailed Design

3.1 Data Collection

Data gathering involves the collection of these datasets. The data is collected from the UCI Machine Learning Repository Bike Sharing Dataset and other relevant sources. The collected data serves as the foundation for building the prediction model

3.2 Data Requirements

The data in this study are shared bike riding records and bike station status, as well as local weather information for that period. The data on the operation of bikes contained information on the time of rental, return time, and place of rental. For the purpose of analyzing daily usage, the data were classified on a daily basis and the daily rental volume was set as a class.

Table 1. Bike share metadata

Fields	Description
Season	1: winter, 2: spring, 3: summer, 4: fall
year	0: 2011, 1:2012
month	1 to 12
hour	0 to 23
holiday	1 = Holiday, 0 = Not a Holiday
weekday	day of the week
working day	1 = Neither a weekend nor holiday, 0 = Either a weekend or a holiday
weather	1 = Clear, Few clouds, Partly cloudy, Partly cloudy 2 = Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist 3 = Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds 4 = Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
temp	Normalized temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-8, t_max=+39 (only in hourly scale)
atemp	Normalized feeling temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-16, t_max=+50 (only in hourly scale)
humidity	Normalized humidity. The values are divided to 100 (max)
wind speed	Normalized wind speed. The values are divided to 67 (max)
casual	Number of non-registered user rentals initiated
registered	Number of registered user rentals initiated
count	Number of total rentals (casual + registered)

3.3 Tool Used

Here are the technical requirements for the bike share prediction model, structured in a detailed and organized manner:

Programming Language and Frameworks

- **Python:** Used as the primary programming language due to its simplicity and the extensive range of libraries available.
- **Libraries**:
  - **NumPy**: For numerical computations.
  - **Pandas**: For data manipulation and analysis.
  - **Scikit-learn**: For building and evaluating machine learning models.
  - **Alternative Modules**: Other relevant modules for specific tasks as needed.

Integrated Development Environment (IDE)

- **Visual Studio Code (VSCode)**: Used as the primary IDE for its versatility and support for Python development.

Data Visualization

- **Seaborn**: For creating attractive and informative statistical graphics.
- **Matplotlib**: Components of Matplotlib are used for creating static, animated, and interactive visualizations in Python.

Front-End Development

- **Flask**: For creating the web application that will serve the prediction model.
- **Stream lit**: For building and deploying the interactive web application.

Version Control and Management

- **GitHub**: Employed for version management and control to track changes, collaborate with team members, and maintain the project's history.

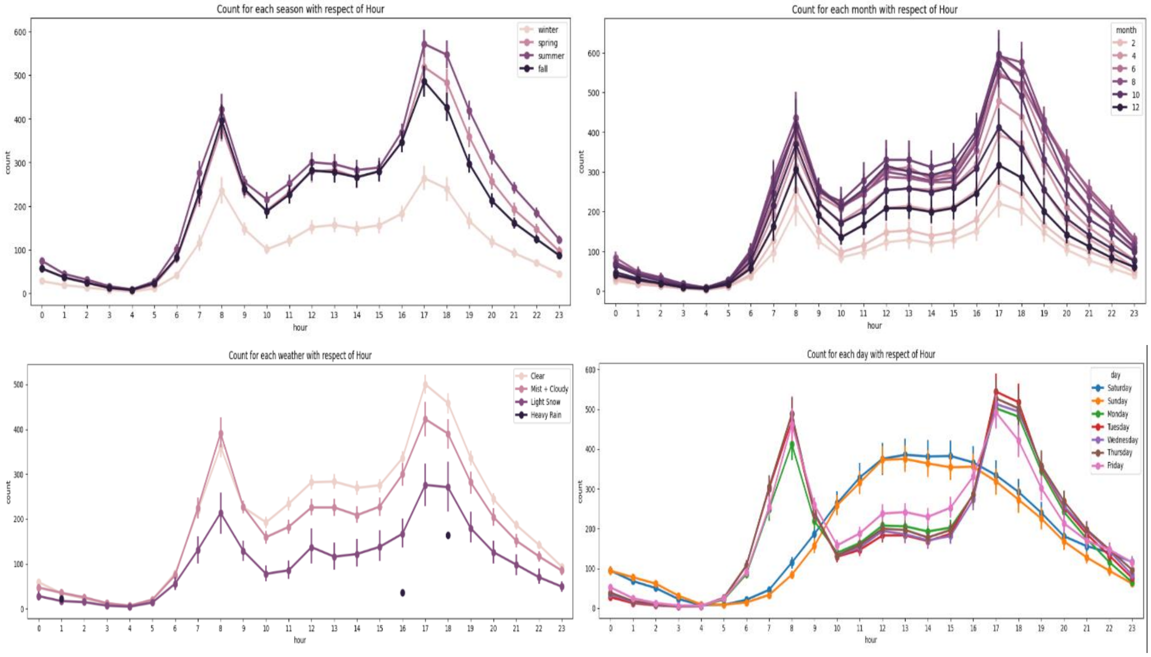
Deployment

- **Stream lit**: Used for the deployment of the web application, allowing for easy sharing and interaction with the model.

3.4 Data Exploration:

Analyzing the collected data involves understanding its structure, patterns, and anomalies. This process includes using descriptive statistics, visualizations, and correlations to gain insights into the data. The findings are summarized below in Figure 2. Bike share demand varies significantly by day of the week and weather conditions. Weekday commuting patterns are evident with high demand during peak hours, while weekends see more consistent usage throughout the day. Clear weather conditions encourage the highest bike share usage, while adverse weather conditions like heavy rain and light snow significantly reduce demand.

Figure 2. Exploratory data analysis summary



3.5 Data Preprocessing:

Cleaning the data involves handling missing values, outliers, and inconsistencies. This includes imputing missing values, removing duplicates, and normalizing data to ensure consistency and reliability. Additionally, data cleaning may involve filtering irrelevant data, correcting errors, and transforming data types. This step is essential to improve the accuracy and performance of the machine learning models.

3.6 Feature Engineering:

Feature engineering transforms raw data into meaningful features for the model. Key techniques include:

- **Time-based Features:** Hour, day of the week, month, and holiday indicators.
- **Weather Conditions:** Temperature, humidity, wind speed, and precipitation levels.

3.7 Model Training:

Training machine learning models using the processed data. Models used include:

- Linear Regression

- Lasso
- Ridge
- K-Neighbors Regressor
- Decision Tree
- Random Forest Regressor
- AdaBoost Regressor

3.8 Model Evaluation Metrics

The following metrics were used to evaluate model performance:

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- R-squared (R2)
- Root Mean Squared Error (RMSE)

Each model's performance was documented in terms of these metrics for both the training and testing datasets.

3.9 Creating Web Application:

Developing a web interface to interact with the prediction system. This setup enables users to input data, receive predictions, and visualize results through a user-friendly interface.

3.10 Deploying on Streamlit:

Deploying the web application using Streamlit for an interactive user interface.

3.11 Storing Input and Output in DB:

Setting up a connection to a database to store and retrieve user data and predictions involves using DataStax to manage input and output data efficiently. Saving user inputs and prediction results in the database for future reference or analysis requires inserting records into a database table every time a prediction is made

3.12 Challenges and Solutions

Challenges:

- Data Quality: Inconsistent and missing data required extensive cleaning and preprocessing.
- Feature Selection: Identifying the most impactful features from a large dataset was complex.
- Real-time Adaptation: Ensuring the model could adapt to real-time data changes posed significant challenges.

Solutions:

- Automated Data Cleaning: Developed scripts to automate the cleaning process.
- Feature Engineering Techniques: Applied advanced feature engineering to enhance model performance.
- Adaptive Learning: Implemented mechanisms for the model to update and adapt to new data.

3.13 Future Work

- Enhanced Real-time Adaptation: Further improve the model's ability to adapt to real-time changes.
- Additional Data Sources: Incorporate more diverse data sources such as social events and road construction information.
- User Feedback Integration: Collect and integrate user feedback to continually improve the system.

3.14 Conclusion

The bike share prediction system project successfully developed a model to forecast bike share demand. By leveraging historical data, weather conditions, and traffic accident data, the model achieved high accuracy and real-time adaptability. The deployment of the model into a web application provides a practical tool for city planners and bike share operators to optimize bike distribution and enhance user experience.

4. Frequently Asked Questions (FAQs)

Q1: What is the primary objective of the bike share prediction system?

A1: The primary objective is to develop a machine learning model that accurately predicts bike share demand in a given city. This helps optimize bike distribution, reduce waiting times, and improve overall user satisfaction.

Q2: What datasets were used in this project?

A2: The project utilized two main datasets:

- **Bike Share Data:** Historical data on bike rentals, including start and end times, durations, and user types.
- **Weather Data:** Information on weather conditions such as temperature, humidity, wind speed, and precipitation.

Q3: How was data processed and cleaned?

A3: Data processing involved handling missing values, correcting erroneous entries, standardizing formats, and merging datasets based on time and location. Automated scripts were developed for these tasks to ensure consistency and efficiency.

Q4: What feature engineering techniques were used?

A4: Key feature engineering techniques included:

- **Time-based Features:** Extracting hour, day of the week, month, and holiday indicators.
- **Weather Conditions:** Incorporating temperature, humidity, wind speed, and precipitation levels.

Q5: Which machine learning algorithms were evaluated?

A5: The following regression algorithms were evaluated:

- Linear Regression
- Lasso
- Ridge
- K-Neighbors Regressor
- Decision Tree
- Random Forest Regressor
- AdaBoost Regressor

Q6: What metrics were used to evaluate model performance?

A6: The model performance was evaluated using the following metrics:

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- R-squared (R2)
- Root Mean Squared Error (RMSE)

Q7: How was the model implemented and deployed?

A7: The chosen model was implemented and deployed using Flask and Streamlit to create a web application. This application allows users to input data and receive real-time predictions on bike share demand.

Q8: What challenges were faced during the project, and how were they addressed?

A8: Challenges:

- **Data Quality:** Inconsistent and missing data.
- **Feature Selection:** Identifying impactful features.
- **Real-time Adaptation:** Ensuring the model adapts to real-time data changes.

Solutions:

- **Automated Data Cleaning:** Scripts for automated cleaning.
- **Advanced Feature Engineering:** Enhanced model performance through robust feature engineering.
- **Adaptive Learning:** Mechanisms for real-time model updates.

Q9: What are the future plans for this project?

A9: Future plans include enhancing real-time adaptation, incorporating additional data sources (such as social events and road construction information), and integrating user feedback to continually improve the system.

Q10: How can the bike share prediction system benefit city planners and bike share operators?

A10: The system provides accurate demand forecasts, helping city planners and bike share operators optimize bike distribution, minimize user wait times, and improve overall service efficiency. This leads to better resource allocation and enhanced user satisfaction.

Q 11: What are the different stages of deployment?

A11: First, the scripts are stored on GitHub as a storage interface.

- The model is first tested in the local environment.

- After successful testing, it is deployed on Streamlit.