Citi Bike Rentals – Analytics and Forecasting

1. Abstract

Our project aims to develop a scalable solution for time series forecasting on Citi Bike dataset using Apache Spark and Facebook Prophet. Time series forecasting is crucial in various domains such as finance, retail, and energy management. However, traditional forecasting methods often struggle to handle large volumes of data efficiently. By leveraging the distributed computing capabilities of Apache Spark and the time series forecasting capabilities of Facebook Prophet, we aim to create a system that can produce scalable analytics and accurate predictions for Citi Bike rental systems. We will address key inquiries posed by Citi Bike such as Where do Citi Bikers ride? When do they ride? How far do they go? Which stations are most popular? What days of the week are most rides taken on?

2. Introduction

Time series forecasting plays a vital role in decision-making processes across industries. The increasing popularity of bike-sharing programs in urban areas has led to a need for efficient management of resources within these systems. Understanding usage patterns and accurately predicting demand can aid in optimizing fleet distribution, station placement, and infrastructure planning. However, existing forecasting methods may not scale well with increasing data volumes, leading to longer processing times and decreased accuracy. Our project addresses this challenge by combining the strengths of Facebook Prophet, a powerful forecasting library, with Apache Spark, a distributed computing framework. By distributing the computational workload across a cluster of machines, we aim to provide a robust forecasting model that provides accurate predictions for Citi Bike rentals.

3. Methodology

We plan to approach the problem by first gathering historical Citi Bike rental data and preprocessing it using Apache Spark. This involves cleaning the data, handling missing values, and feature engineering. We will then utilize the distributed computing capabilities of Apache Spark to perform exploratory data analysis and feature extraction efficiently. Next, we will employ Facebook's Prophet library to develop time series forecasting models for predicting bike rental demand. We will train and validate the models using historical data and evaluate their performance.

4. Evaluation

We will evaluate our project based on its ability to generate accurate forecasts in a scalable manner. The primary success metric will be the Mean Absolute Percentage Error (MAPE) of the forecasts compared to actual values. Other metrics such as RMSE (Root Mean Square Error), MAE (Mean Absolute Error) will be used for assessing performance in prediction. Additionally, we will involve two parallel approaches: one employing Apache Spark for data preprocessing, analysis, and model training, and the other utilizing traditional methods without Spark. Through this comparative analysis, we seek to understand the extent to which Spark enhances the scalability and computational efficiency of our forecasting process.

5. Data

For our project, we will utilize the Citi Bike trip data provided by the New York City Bike Share (NYCBS) program. This dataset includes comprehensive information on Citi Bike trips, enabling us to answer various questions related to rider behavior and usage patterns. The dataset contains the following fields: Ride ID, Rideable type, Started at, Ended at, Start station name, Start station ID, End station name, End station ID, Start latitude, Start longitude, End latitude, End Longitude, Member or casual ride. With this rich dataset, we can analyze various aspects of Citi Bike usage, including rider destinations, ride durations, popular stations, time-of-day trends, and the breakdown between member and casual rides. This data will serve as the foundation for our analytics and forecasting project, allowing us to derive valuable insights and develop predictive models to optimize bike-sharing operations.
Dataset Link: Citi Bike System Data | Citi Bike NYC | Citi Bike NYC

6. Task List

1. Data collection and preprocessing
2. Exploratory data analysis and feature extraction:
3. Model Development
4. Evaluation
5. Performance optimization and scalability testing