NYC Yellow Taxi Trip Record Data Analysis

Shantanu Chaudhari
Mehta Family School of Data Science
and Artificial Intelligence
Indian Institute of Technology
Guwahati, *India*
c.ujwal@iitg.ac.in

***Abstract*— The advent of digital technologies has catalyzed a data-driven transformation across various sectors, and the transportation industry is no exception. This study delves into the vast dataset of New York City (NYC) Yellow Taxi trip records to uncover intricate patterns, trends, and behavioral insights. Leveraging advanced data analytics and visualization techniques**

Keywords— Big Data Analytics, Transportation, Data Visualization, Taxi.

I. Introduction

The New York City (NYC) taxi industry stands as an emblematic facet of urban transportation, catering to millions of residents and visitors alike. With the omnipresence of yellow cabs navigating the bustling streets, the demand for efficient and accurate fare prediction systems has become increasingly paramount. In this era of data-driven decision-making, the fusion of advanced analytics and machine learning offers an unprecedented opportunity to revolutionize the taxi fare estimation process.

This report delves into a comprehensive NYC Taxi Fare Prediction project, aiming to leverage cutting-edge technology to enhance the precision and reliability of fare predictions. The complexities of urban taxi systems, coupled with the dynamic nature of NYC's traffic and demand patterns, necessitate a sophisticated approach to model development. Through the amalgamation of historical taxi trip data, environmental factors, and temporal considerations, our project aspires to yield a predictive model capable of offering commuters and taxi service providers accurate fare estimates in real-time.

As we navigate through the various stages of this project, from data collection and preprocessing to model selection and evaluation, the overarching objective remains clear: to develop a robust and scalable fare prediction system that not only meets the immediate needs of taxi passengers but also contributes to the optimization of taxi fleet operations. By addressing the inherent challenges posed by the NYC taxi landscape, we anticipate that the outcomes of this project will not only benefit the end-users but also offer insights and methodologies applicable to broader transportation and urban planning domains.

II. Problem Statement

The NYC taxi system grapples with the challenge of inaccurate fare predictions due to the dynamic nature of traffic, varying demand, and environmental factors. Current static pricing models fall short in providing real-time, precise estimates, leading to inconveniences for passengers and operational inefficiencies for service providers. This project aims to address this issue by developing an advanced predictive model, incorporating historical trip data, environmental variables, and temporal factors. The goal is to optimize fare predictions, enhancing user satisfaction and operational efficiency in NYC's taxi services while potentially offering broader applications in urban transportation.

III. Architecture

A. Model Training

The data is provided in PARQUET format which makes running queries on data easier. So, the data will be stored as a single or multiple Apache PARQUET files which will be later accessed by the analytics tools down the pipeline.

A model training script collects the data and performs operations on it. Initially unnecessary columns are dropped and requires ones are cleaned based on past Exploratory Data Analysis. Features such as extreme travel time, negative fares, extreme fares etc are dropped. End result is a clean DataFrame.

The Data is then fed to a pipeline consisting of Indexer, Assembler and finally a Random Forest based Model. The Indexer automatically finds categorical features and indexes them. Assembler assembles all the features in a single vector. And lastly the model gets trained on the Assembled data.

The model gets saved to a location in the server and later gets accessed by the Django API.

B. Django Prediction API

The Django API does several things, which are:

- Finding the specific borough associated with the Geographic location
- Getting the estimated distance and time for travelling between the pickup and drop-off location using DistanceMatrix.ai API
- Predicting the fare using the trained model in the previous step

The API receives latitude and longitude for the pickup and drop-off locations. The API then calculates the other required features namely (distance, travel time, location ids, year, hour, month and day of week). The Distance and time are calculated using the distancematrix.ai api. The borough associated with the latitude and longitude is calculated from the shapefile provided by the NYC TLC.
The API then predicts the fare using the random forest regressor model trained in the previous step.

C. Frontend APP

The frontend app is meant to get latitude and longitude information from the user for pickup and drop-off locations and send it to the backend server. The backend server responds with the predicted fare for the same.

Currently it only has text fields as input for latitude and longitude. It was aimed to include google maps to

pick the location on map and extract the location from it. But due to paid nature of google maps api and complexity of alternative solution, it was put on halt.

IV. Demo

The demonstration for the project can be found on the GitHub link provided in the course.

https://github.com/chiranjibsuriitg/DA331fall23_210150023

V. Results

The implementation of a Django backend for training the predictive model and handling real-time predictions has yielded promising outcomes. The model, trained on a comprehensive dataset encompassing historical taxi trip data, environmental variables, and temporal considerations, demonstrates a notable improvement in fare prediction accuracy compared to traditional static pricing models.

The frontend application, designed to interact seamlessly with the Django backend, has proven effective in providing the necessary input data for predictions. User inputs, including pick-up and drop-off locations, time of day, and weather conditions, are seamlessly transmitted to the backend for processing. The integration between the frontend and backend components ensures a user-friendly experience while facilitating the flow of relevant data to enhance the model's predictive capabilities.

The predictive model, fine-tuned through the Django backend, showcases its adaptability to dynamic factors such as fluctuations in demand and changes in environmental conditions. Real-time predictions generated by the model align closely with actual fare outcomes, offering a reliable tool for both taxi service providers and passengers seeking accurate estimates for their journeys.

VI. Conclusions

In conclusion, the integration of a Django backend and frontend application has proven to be a robust solution for addressing the challenges associated with inaccurate fare predictions in the NYC taxi industry. The predictive model, trained on diverse and dynamic datasets, showcases a significant enhancement in accuracy, offering a viable alternative to static pricing models.

The seamless interaction between the frontend and backend components not only ensures a smooth user experience but also establishes a scalable framework for future enhancements and updates. The adaptability of the model to real-time variables positions it as a valuable tool for taxi service providers to optimize fleet operations and for passengers to receive accurate fare estimates.

This project not only contributes to the immediate needs of the NYC taxi industry but also serves as a testament to the potential of advanced analytics and machine learning in solving complex challenges within urban transportation. The collaborative efforts between data scientists, developers, and stakeholders have resulted in a practical and effective solution that can be extended and adapted to address similar issues in other urban transit systems.