Taxi Fare Predictors

Introduction:

To create predictions, the industry today significantly relies on data analytics. These forecasts result in profitable business strategies that heavily rely on machine learning. Before matching a consumer to a driver, popular taxi services like Uber and Lyft give their users an estimate of the cost of the taxi. Using the public dataset made available by the NYC Taxi and Limousine Commission (NYC-TLC), we attempt to offer a comparable solution. The goal is to do feature engineering on massive amounts of data collected in NYC-TLC's open data repository, train a prediction model utilizing that data, and then deploy that model. The crucial concept is to comprehend and put into practice a data analytics pipeline, which serves as the cornerstone of data processing in modern software engineering.

Motivation:

Some of the main impetus for starting this project are as follows:
Prediction of Taxi Fares: A crucial component of this project is the accurate estimation of taxi fare. Such forecasts are useful for both drivers and passengers, who may strategically plan their trips to maximize revenues and make more informed financial decisions. We can improve the overall taxi experience for both parties by anticipating fare amounts.
Surge Prediction: Another important goal is to anticipate regions with strong demand and probable spikes in taxi orders. Drivers may strategically position themselves thanks to these predictive capabilities, which decreases passenger wait times and boosts the effectiveness of taxi services as a whole. This has the potential to greatly increase consumer satisfaction.
Identification of Hot Spots: We can identify hotspots where demand for taxi services is regularly strong by examining the geographic distribution of taxi service requests. Finding these hotspots is essential for making wise business decisions, such as where to place more drivers or focus marketing initiatives to draw in more riders.
The creation of a system that can skillfully handle the enormous volumes of data generated by the NYC TLC on a daily basis is the primary driving force behind this project. We are dedicated to taking on the scalability question head-on, in contrast to conventional machine learning methods, which frequently struggle to maintain effectiveness as data quantities increase. Our goal is to build an infrastructure that can successfully handle the current data deluge while also holding up against potential future development.

Literature Review:

The paper [Elmi21] presents a novel deep learning architecture called TRES-Net for the prediction of taxi fares by addressing spatial-temporal dependencies. TRES-Net combines Residual Networks (ResNet) for spatial information and Bi-directional Long Short-Term Memory (Bi-LSTM) for temporal information. It constructs a matrix of similar trips based on road network structure and uses a lookup operation to embed spatial features from similar trips. A Bi-LSTM is employed to capture time-series patterns, and the model incorporates a periodically shifted attention mechanism to handle long-term periodic dependencies. TRES-Net also considers external factors like holidays and events, capturing similarity between days of the week and hours of the day. The model combines ResNet and Bi-LSTM, aiming to balance memorization and representation abilities. In terms of performance, TRES-Net outperforms existing models in terms of Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) on real datasets. It achieves better results when compared to nine benchmark methods across various prediction intervals. Future work could focus on real-time model adaptation to changing traffic patterns without retraining from scratch.

Taxi Fare Predictors

This paper [Bagal23] discusses the implementation of a Deep Neural Network (DNN) for predicting taxi and cab fares, focusing on the design and implementation of the algorithm. DNNs are commonly used for image recognition and classification, but they can also be applied to numerical data such as time series data. The paper emphasizes the adaptability of DNNs in handling various data types. The algorithm, termed "Parallel-DNN," is outlined through pseudo-code. It involves steps like feature scaling, splitting data into training and testing sets, and training and testing the DNN model. The dataset used for this project is obtained from Kaggle and contains information related to taxi trips, including fare amounts, timestamps, passenger counts, and location coordinates. The paper highlights the advantages of using DNNs, such as their high accuracy and the ability to handle large datasets with minimal pre-processing. The results of implementing the DNN are compared to other algorithms, including Linear Regression, Random Forest, Decision Tree, Gradient Boosting, and XGBoost Regressor. The DNN outperforms these algorithms in terms of Root Mean Squared Error (RMSE) and Mean Squared Error (MSE). This paper demonstrates the effectiveness of DNNs in handling numerical data and offers promising results in terms of prediction accuracy. It suggests that DNNs can be valuable tools for enhancing service quality and pricing strategies in the taxi industry.

This paper [Howard18] explores the development of a smart transportation data pipeline using IoT data from the New York City Taxi & Limousine Commission. The authors conduct experiments on various machine learning algorithms, both supervised and unsupervised, to assess their prediction accuracy and computational performance on both commodity computers and distributed systems. Their chosen technologies include Amazon S3, EC2, EMR, MongoDB, Sagemaker and Spark. Notably, the Random Forest Regressors achieve a Root Mean Squared Error (RMSE) of 0.22, a result comparable to Kaggle competition winners. Surprisingly, Logistic Regression outperforms more complex algorithms in terms of training time and accuracy in some instances, underscoring the importance of algorithm selection and data size. The paper's metrics for result measurements include RMSE for prediction accuracy and training times for various machine learning algorithms, particularly comparing Logistic Regression to Random Forest and Gradient Boosted Tree classifiers. This study provides valuable insights into the optimal choice of algorithms and infrastructure for transportation data analysis in distributed systems, making it a pertinent resource for researchers in the field.

This research paper [Ismaeil22] focuses on the application of a decision tree classification model to predict payment types in New York City (NYC) taxi trips. The authors utilize a dataset provided by the NYC Taxi and Limousine Commission, which includes detailed trip records with information on vendors, passenger counts, pickup/drop-off locations, timestamps, and payment types. The goal of the study is to assess the accuracy of payment type prediction using the decision tree classification algorithm within the Apache Spark framework.The research demonstrates a high level of accuracy, exceeding 96%, for predicting payment types using the decision tree model. The authors discuss various evaluation metrics, including accuracy, test error, weighted precision, weighted recall, and weighted F-measure. They also investigate the impact of different splitting criteria and node impurity measures (Gini impurity and Entropy) on the model's performance. Overall, this paper offers valuable insights into the application of machine learning, specifically decision tree classification, to real-world transportation data. The findings suggest that this approach can be extended to predict other aspects of taxi trips, such as passenger count and peak-hour surcharges, and potentially be applied to similar car services in different regions.

Taxi Fare Predictors

This research paper [Sun16] analyzes a vast dataset of NYC taxis with Big Data technologies. The research utilizes MapReduce and Hive to understand the patterns and make a prediction on taxi networks. To begin, taxi trip data sourced from the NYC Taxi & Limousine Commission website is transferred into the Hadoop cluster, initiating the data analysis process. This data is then placed within the Hadoop Distributed File System (HDFS), allowing for efficient processing through a suite of tools, including Hadoop MapReduce, Hive, HBase, Pig, and Spark.The primary analysis of the dataset involves MapReduce and Hive, with intermediate results being checked for discrepancies. Subsequently, these intermediate results are processed, confirmed, and graphed to visually represent data relationships and patterns. The architecture treats every taxi as a moving sensor within the NYC taxi system. By combining the historical data gathered from these taxis with a smartphone application, it offers valuable services to taxi riders, drivers, and taxi companies.

The research paper [Yazici13] utilizes a large dataset of taxi trips to analyze the decision-making process of New York City taxi drivers, aiming to propose strategies for improving access to and passenger satisfaction at John F. Kennedy (JFK) Airport. A one-month subset of taxi GPS data supplied by the New York City Taxi and Limousine Commission (TLC) is utilized. The dataset comprises essential information, including the latitude and longitude coordinates of pick-up and drop-off locations, along with timestamps. They identified sequential taxi trips and used logistic regression to model whether drivers chose to pick up passengers at the airport or search for customers after each trip. The results of their model confirmed various issues raised by industry stakeholders. Additionally, this methodology can be applied in other locations with access to similar taxi trip datasets.

Design of the System:

Amazon S3 - The NYC-TLC dataset is used for the project which is publicly available in S3 bucket. So we fetch the data from this bucket and store it in our local S3 bucket.
Spark cluster on EMR - This is used for data preprocessing and feature engineering of the large dataset fetched from the Amazon S3 bucket. Then the cleaned data is again stored to the S3 bucket.
Amazon Sagemaker - The Amazon Sagemaker is used to train and test the model using different machine learning techniques. Then it will be stored in the form of a pickled model.
Web Application - Then we are planning to make a web application to show visualizations and predict the actual taxi fare.

Implementation:

In our implementation, we aim to utilize various resources, including a public dataset hosted on NYC-TLC S3, AWS EMR, Apache Spark, AWS Sagemaker, Pandas, Python, and Flask. Our approach is to adapt different technologies from different papers and implement them in our project. We will streamline our architecture, focusing on batch processing. Our data ingestion will directly integrate with an EMR cluster running Spark [Howard18], allowing for efficient data preprocessing, including data cleansing and feature engineering, such as introducing zip codes for pickup and dropoff locations and creating trip frequency features for location-based insights.

A key advantage of our methodology lies in the efficient parallelization [Bagal23] and memory-based processing. We aim to overcome the bottleneck observed in previous methods that involved excessive disk-based I/O. By loading input files into memory as Pandas dataframes and moving the process toward data, we expect a significant improvement in performance. This approach will substantially reduce processing time, making it more efficient for real-world applications. Our system will store processed data back in S3, and we will employ AWS Sagemaker [Howard18] for model training, including hyperparameter optimization and cross-validation.

Moreover, we aim to lead to a user-friendly Python Flask web application, providing a seamless interface for users to query the model using parameters like source and destination addresses and passenger count, with the additional benefit of Google Maps API integration. By focusing on efficiency, scalability, and user-friendliness, our methodology aims to outperform the previous six papers in terms of practicality and real-world applicability.

Challenges Overcome:

Scalability: One of the foremost challenges in big data projects is ensuring scalability as datasets grow. We are committed to overcoming this challenge by effectively handling the vast volumes of data generated by the NYC Taxi and Limousine Commission on a daily basis. We achieve this through the utilization of AWS EMR (Elastic MapReduce) and Apache Spark, allowing for efficient parallel processing. By harnessing cloud-based clusters, our system seamlessly scales to handle the ever-increasing data quantities, ensuring its continued effectiveness.

Data Preprocessing: Data cleansing and feature engineering are pivotal in developing accurate prediction models. Our project addresses the challenge of data preprocessing by integrating with an EMR cluster running Spark. This approach enables efficient data cleansing and feature engineering, including the introduction of zip codes for pickup and dropoff locations, and the creation of trip frequency features. Streamlining these processes ensures our data is well-prepared for modeling**.**

Model Training and Hyperparameter Optimization: Training machine learning models on large datasets can be computationally intensive. To address this, our system leverages AWS Sagemaker for model training, hyperparameter optimization, and cross-validation. Sagemaker offers a scalable and cloud-based infrastructure for training models, ensuring that this resource-intensive task is handled efficiently.

Conclusion:

In today's industry, data analytics plays a pivotal role in making predictions that are essential for formulating profitable business strategies, particularly in the context of machine learning. This is evident in the operations of major taxi services like Uber and Lyft, where they provide users with estimates of taxi fares before matching them with a driver. We aim to create a predictive solution by performing feature engineering on NYC-TLC’s and deploying the prediction model to make it accessible for users to obtain fare predictions. In essence, the project aims to harness the power of data analytics to provide users with more transparent and informed cost estimates for their taxi rides, much like popular ride-sharing services do.

Taxi Fare Predictors

References:

[1] S. Elmi and T. Kian-Lee, "Learned Taxi Fare for real-life trip trajectories via Temporal ResNet Exploration," in MobiQuitous '20: MobiQuitous 2020 - 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, December 2020, pp. 86–95, https://doi.org/10.1145/3448891.3448893, Published: 09 August 2021.

[2] N. M. Bagal, M. D. Gabhane, and C. V. Mahamuni, "Rideshare Transportation Fare Prediction using Deep Neural Networks," in 2023 International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 11-12 May 2023, doi: 10.1109/ICDT57929.2023.10150947.

[3] A. J. Howard, T. Lee, S. Mahar, P. Intrevado and D. Myung-Kyung Woodbridge, "Distributed Data Analytics Framework for Smart Transportation," 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Exeter, UK, 2018, pp. 1374-1380, doi: 10.1109/HPCC/SmartCity/DSS.2018.00227.

[4] H. Ismaeil, M. A. Abdel-Fattah, and S. Kholeif, "Using Decision Tree Classification Models to Predict Payment Type in NYC Yellow Taxi," in International Journal of Computer Science and Information Security (IJCSIS), vol. 20, no. 2, pp. 37-42, Feb. 2022. doi: 10.5281/zenodo.6379884.

[5] H. Sun and S. McIntosh, "Big Data Mobile Services for New York City Taxi Riders and Drivers," 2016 IEEE International Conference on Mobile Services (MS), San Francisco, CA, USA, 2016, pp. 57-64, doi: 10.1109/MobServ.2016.19.

[6] M. A. Yazici, C. Kamga and A. Singhal, "A big data driven model for taxi drivers' airport pick-up decisions in New York City," 2013 IEEE International Conference on Big Data, Silicon Valley, CA, USA, 2013, pp. 37-44, doi: 10.1109/BigData.2013.6691775.