American University of Armenia
Fall 2023

DS227A: Business Analytics

Predictive Modeling for Cab Ride Pricing

Professor:
Nelli Muradyan

Authors:
Inna Krmoyan
Gor Mkrtchyan

Abstract

Our project aims to develop a robust predictive model for forecasting cab ride prices to include the recommendation of the most convenient cab company at a given moment. The objective is to provide users with a comprehensive and personalized experience by combining accurate fare predictions with real-time assessments of cab availability and pricing competitiveness.

The project methodology involves integrating historical pricing data with real-time data streams from two cab service providers that are Uber and Lyft. Machine learning algorithms will be employed to not only forecast ride prices based on various factors but also to analyze ongoing trends in cab availability, and pricing strategies of those companies. The model will consider user preferences, such as preferred cab providers and route choices, to tailor recommendations accordingly.

The anticipated outcomes include a holistic platform that seamlessly integrates predictive pricing and provider recommendation functionalities. The implementation of such a model has the potential to revolutionize the user experience in the ride-sharing industry by offering users not only accurate fare estimates but also intelligent suggestions on the most convenient and cost-effective cab company for their specific journey.

This project's contributions extend to both users and cab service providers, optimizing user satisfaction and enhancing operational efficiency. Furthermore, the insights derived from the analysis of real-time data could inform strategic decisions for cab companies, enabling them to adapt their pricing and service offerings based on current market dynamics.

Problem/Hypothesis Statement

In the rapidly evolving landscape of ride-sharing services, the variability in cab ride prices presents a significant challenge for users seeking reliable and transparent fare estimates. The fluctuating nature of pricing on platforms like Uber and Lyft, driven by factors such as demand, time of day, and weather conditions, underscores the need for a predictive model that not only anticipates fare costs accurately but also recommends the most convenient cab service at any given moment.

Implementing an advanced predictive modeling approach, incorporating historical ride data from Uber and Lyft, as well as weather-related information, will enable the development of a dynamic pricing model. This model aims to provide users with precise fare estimates based on contextual factors while simultaneously offering intelligent recommendations on the most convenient cab service. By analyzing patterns in demand, time-sensitive surges, and the impact of weather conditions, we hypothesize that our predictive model can enhance user experience, foster user trust, and optimize the operational efficiency of ride-sharing services.

Data

We have found two datasets for our analysis: one comprises cab ride information from industry leaders Uber and Lyft, while the other dataset focuses on comprehensive weather data. The pricing dynamics of Uber and Lyft differ significantly from fixed-rate public transport, fluctuating based on the prevailing demand and supply of rides. Identifying the factors influencing this demand is crucial. Intuitively, peak commuting hours, such as around 9 am and 5 pm, are expected to exhibit heightened demand due to individuals traveling to and from work. Additionally, meteorological conditions, including rain or snow, are anticipated to impact ride frequency, as adverse weather conditions often prompt increased reliance on ride-sharing services.

Uber and Lyft dataset:

Distance - distance between source and destination
Cab_type - Uber or Lyft
Time_stamp - epoch time when data was queried
Destination - destination of the ride
Source - the starting point of the ride
Price - price estimate for the ride in USD
Surge_multiplier - the multiplier by which price was increased, default 1
Id - unique identifier
Product_id - uber/lyft identifier for cab-type
Name - Visible type of the cab eg: Uber Pool, UberXL

Weather dataset:
Temp - Temperature in F
Location - Location name
Clouds - Clouds

Pressure - pressure in mb
Rain - rain in inches for the last hr
Time_stamp - epoch time when row data was collected
Humidity - humidity in %
Wind - wind speed in mph

*** *Data Source*

Introduction: Getting to know the datasets

Prior to working on the development of our project's actual model, it is imperative to undertake meticulous data cleaning, manipulation, and dataset integration.
As the datasets that we got were not as messy as expected, here are the steps we undertook during the first stage of our project.

1. Filling Null Values:

Filling the null values in the column “rain” with 0, indicating that there was no recorded rainfall at those specific times.

2. Converting Unix Epoch Timestamps to DateTime:

We converted Unix Epoch timestamps to human-readable DateTime format for both the 'cabs' and 'weather' DataFrames.

3. Merging DataFrames:

Eventually we merged the 'cabs' and 'weather' DataFrames based on common columns ('source' and 'time_stamp').

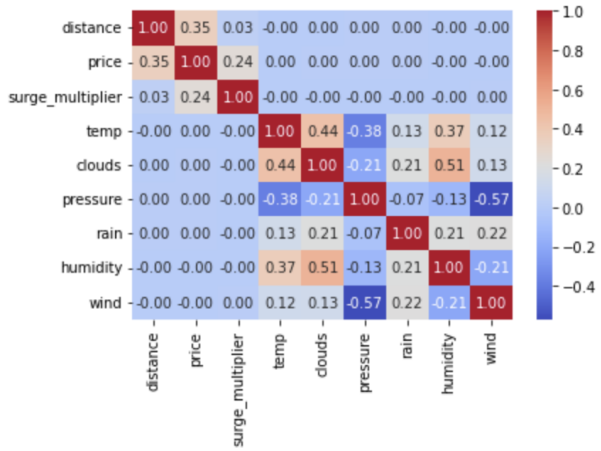


Fig. 1

After examining the correlation matrix above, the given correlations do make a lot of sense as those features are mostly dependant on one another.

	distance	price	surge_multiplier	temp	clouds	pressure	rain	humidity	wind
distance	1.000000	0.345082	0.025830	-0.003099	0.000342	0.003708	0.000789	-0.003740	-0.004251
price	0.345082	1.000000	0.240325	0.000005	0.001290	0.001012	0.000710	-0.001704	-0.001185
surge_multiplier	0.025830	0.240325	1.000000	-0.001635	-0.001996	-0.002912	-0.001067	-0.001253	0.001286
temp	-0.003099	0.000005	-0.001635	1.000000	0.437995	-0.377918	0.133400	0.366563	0.120664
clouds	0.000342	0.001290	-0.001996	0.437995	1.000000	-0.210674	0.210941	0.509777	0.125876
pressure	0.003708	0.001012	-0.002912	-0.377918	-0.210674	1.000000	-0.069215	-0.132083	-0.569770
rain	0.000789	0.000710	-0.001067	0.133400	0.210941	-0.069215	1.000000	0.209892	0.218114
humidity	-0.003740	-0.001704	-0.001253	0.366563	0.509777	-0.132083	0.209892	1.000000	-0.208435
wind	-0.004251	-0.001185	0.001286	0.120664	0.125876	-0.569770	0.218114	-0.208435	1.000000

Table 1

Here is a summary statistics table for our data.

The correlation matrix reveals several key associations among the variables. Distance exhibits a moderate positive correlation with price (0.345), indicating that longer rides generally incur higher costs. The surge_multiplier is moderately positively correlated with price (0.240), suggesting a nuanced relationship between surge multipliers and fare amounts. Temperature shows minimal linear correlation with other variables. Cloud cover demonstrates weak positive correlations with temperature (0.438), pressure (0.211), and humidity (0.510). Pressure displays a moderate negative correlation with temperature (-0.378) and a weak negative correlation with humidity (-0.132). Rain exhibits weak positive correlations with temperature (0.133), cloud cover (0.211), and humidity (0.210). Humidity is moderately positively correlated with temperature (0.367) and cloud cover (0.510), with weak negative correlations with pressure (-0.132) and wind speed (-0.208). Wind speed demonstrates a moderate negative correlation with pressure (-0.570) and a weak negative correlation with humidity (-0.208). These insights provide a nuanced understanding of the interplay between various factors in the context of cab ride pricing.

Modeling approach

Before going on with the models, we started off with feature engineering. As mentioned above, we had two datasets but we merged those two and cleaned the data. Here is what we were left with: **'distance', 'cab_company', 'order_date', 'destination', 'origin', 'price',**

'surge_multiplier', 'order_id', 'car_type', 'temp', 'clouds', 'pressure', 'rain', 'humidity', 'wind', 'car_category', 'hour', 'day_of_week'

New columns were added (**'car_category', 'hour', 'day_of_week'**), as well as the column's **'car_type'** values have been renamed, in such manner:

'Shared': 'Lyft Shared',
'UberPool': 'UberPool',
'Lux': 'Lyft Lux',
'Lux Black': 'Lyft Lux',
'Black': 'Uber Lux',
'Lux Black XL': 'Lyft Lux XL',
'Black SUV': 'Uber Lux XL',
'UberXL': 'Uber XL',
'Lyft XL': 'Lyft XL',
'Lyft': 'Lyft',
'UberX': 'UberX',
'WAV': 'Uber WAV'

This was done to easier distinguish the cars for the further investigations. **'Car_category'** includes values: 'Shared', 'Lux', 'Lux XL', 'XL', 'Standard', 'Wheelchair', this is a broader category for the subcategory car type. **'Hour'** and **'day_of_week'** were extracted from 'Date' column.

We had three models for forecasting and one model for recommendations.

1. ARIMA Model:

- The Autoregressive Integrated Moving Average (ARIMA) model is employed for time-series forecasting. It captures temporal patterns and trends in the data.
- The training data is divided into an 80-20 split, with the ARIMA model fitted to the training set.
- The model is then used to forecast future values on the test set, and the results are plotted alongside the actual values.
- The ARIMA model is effective for capturing autocorrelation and seasonality in the time series.

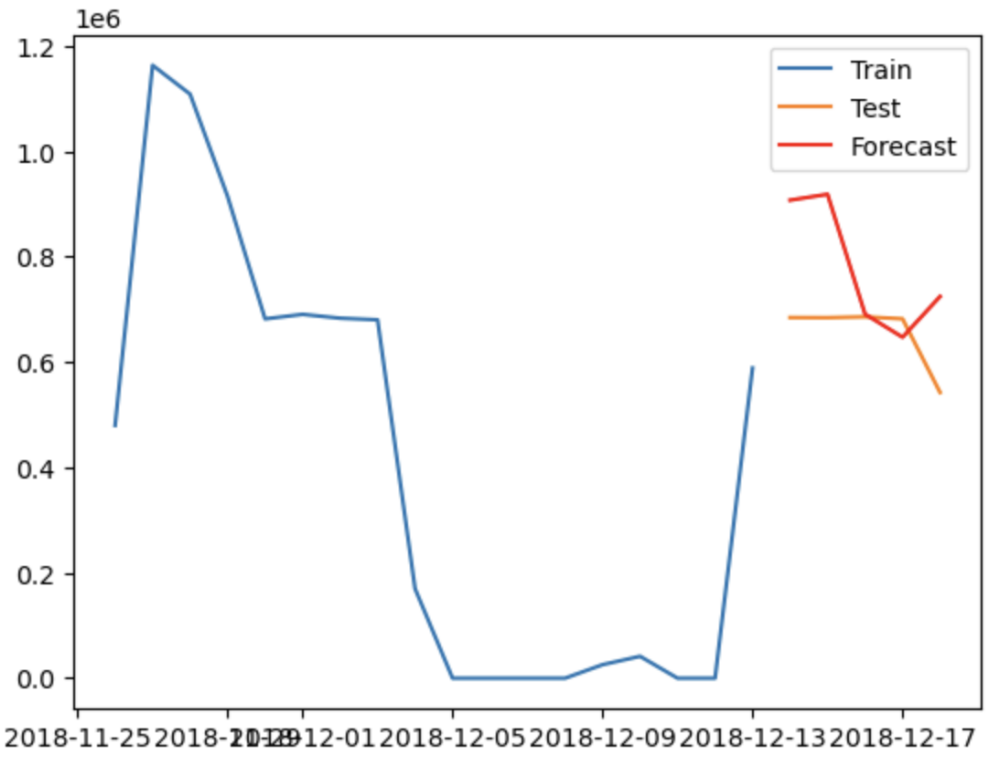


Fig. 2

2. ETS Model (Exponential Smoothing State Space Model):

- The Exponential Smoothing State Space Model is utilized for time-series forecasting with an additive trend and additive seasonality (seasonal period of 7 days).
- Similar to the ARIMA model, it is trained on 80% of the data and tested on the remaining 20%.
- The model's predictions are plotted against the actual values, showing its ability to capture trends and seasonality.

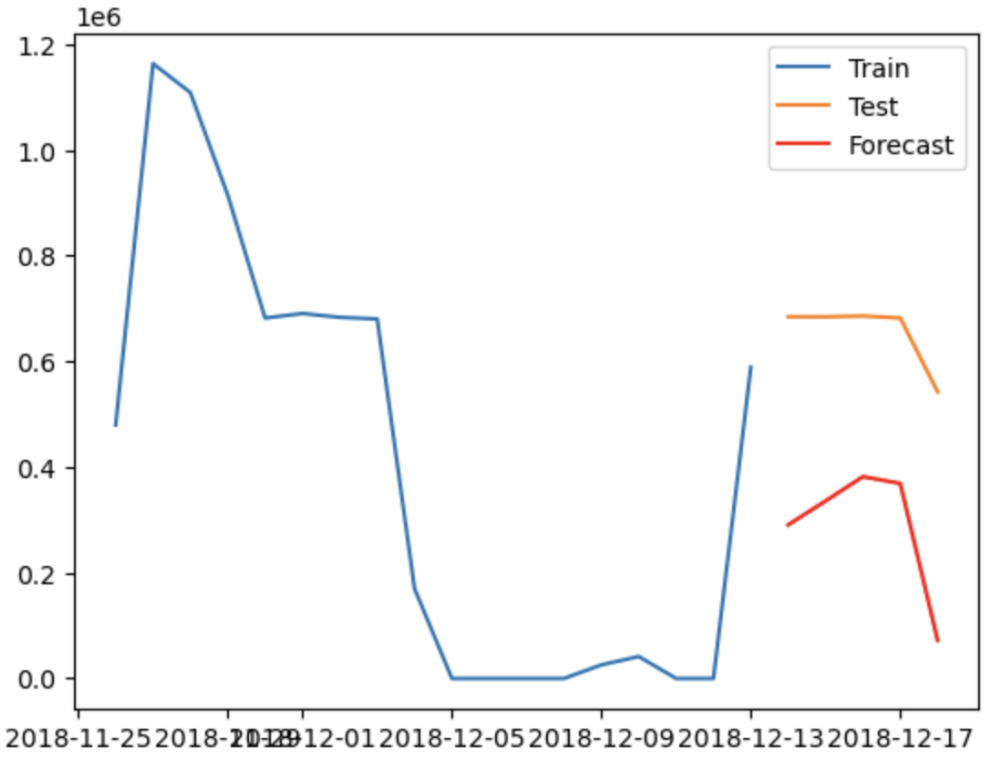


Fig. 3

3. Gaussian Process Regression:

- Gaussian Process Regression (GPR) is employed for non-linear regression, allowing for flexible modeling of complex relationships.
- The training data is used to fit the Gaussian process regression model, and the resulting model is applied to forecast future values on the test set.
- The model's predictions are plotted, including a shaded area representing the standard deviation, providing a measure of uncertainty.
- GPR is valuable for capturing non-linear trends and uncertainties in the time series.

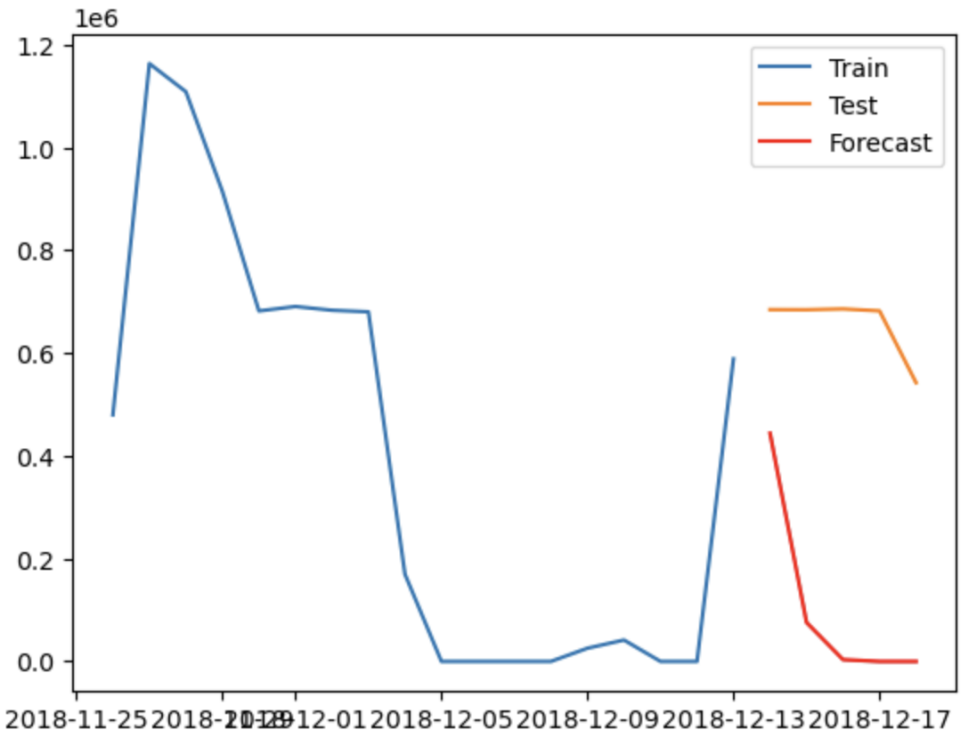


Fig. 4

After doing forecasting on our data via different models, we also decided to create a recommendation system for the cab rides.It was designed to offer ride recommendations based on user inputs such as origin, destination, day of the week, and hour of the day. The function filters the dataset according to these inputs, calculates average prices for each car category and type, and suggests the cheapest option for the specified conditions. If data for the exact hour is not available, the function estimates the daily average and provides recommendations based on that. The script then takes user inputs for their current location, destination, day of the week, and hour of the day and calls the `recommend_ride` function to display personalized ride recommendations.

Here is a step-by-step representation of the model.

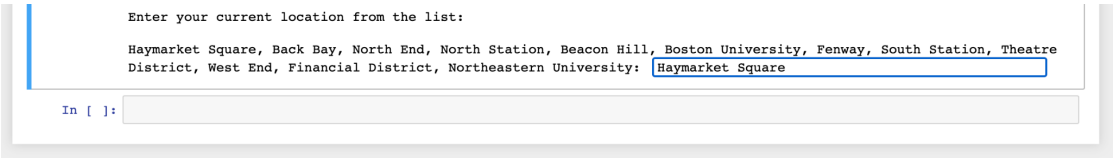
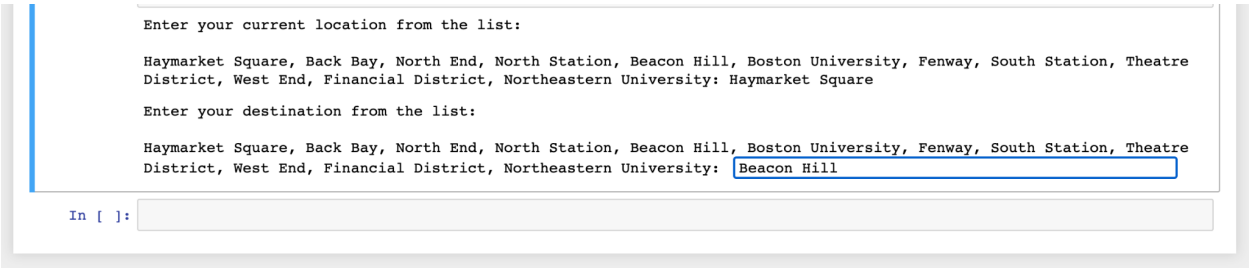


Fig. 5

Asking the user their current location from where they are going to order a cab.



.Fig. 6

Asking the user their destination.

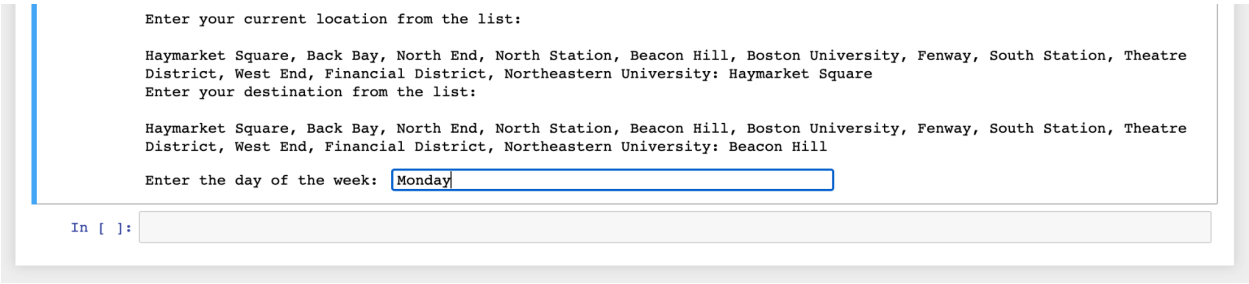


Fig. 7

Asking the user for the desired day of the week they intend to take the cab.

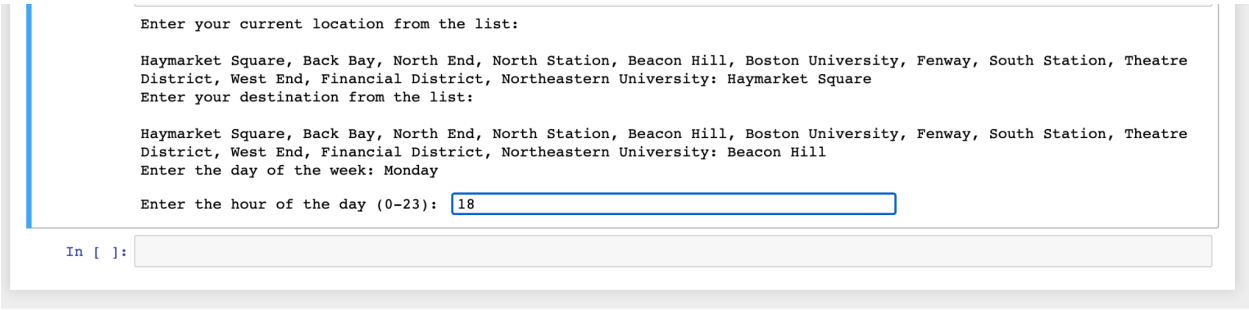


Fig. 8

Asking the user for the hour.

```
Enter your current location from the list:

Haymarket Square, Back Bay, North End, North Station, Beacon Hill, Boston University, Fenway, South Station, Theatre
District, West End, Financial District, Northeastern University: Haymarket Square
Enter your destination from the list:

Haymarket Square, Back Bay, North End, North Station, Beacon Hill, Boston University, Fenway, South Station, Theatre
District, West End, Financial District, Northeastern University: Beacon Hill
Enter the day of the week: Monday
Enter the hour of the day (0-23): 18
car_category: Shared
Lyft Shared: 4.33
UberPool: 8.02
We suggest using Lyft Shared

car_category: Lux
Lyft Lux: 15.20
Uber Lux: 16.29
We suggest using Lyft Lux

car_category: Lux XL
Lyft Lux XL: 26.74
Uber Lux XL: 26.89
We suggest using Lyft Lux XL

car_category: XL
Lyft XL: 11.41
Uber XL: 12.50
We suggest using Lyft XL

car_category: Standard
Lyft: 7.19
UberX: 8.00
We suggest using Lyft

car_category: Wheelchair
Uber WAV: 8.05
We suggest using Uber WAV
```

Fig. 9

The final recommendation output.

Analysis & findings/results

1. ARIMA Model:

- MAE: The average absolute difference between the actual and predicted values is $135,824.88. This indicates that, on average, the model's predictions deviate by this amount from the actual values.
- MSE: The mean squared error is $27,775,449,458.73. This measures the average squared difference between actual and predicted values, giving higher weight to larger errors.
- RMSE: The root mean squared error is $166,659.68. It represents the square root of the MSE and provides an interpretable metric in the same units as the target variable.
- MAPE: The mean absolute percentage error is 21.24%, indicating that, on average, the model's predictions deviate by approximately 21.24% from the actual values.

2. ETS Model (Exponential Smoothing):

- MAE: The average absolute difference for the ETS model is $365,976.69.
- MSE: The mean squared error is $137,647,297,010.91.

- RMSE: The root mean squared error is $371,008.49.
- MAPE: The mean absolute percentage error is 57.07%.

3. Gaussian Process Regression:

- MAE: The average absolute difference for the Gaussian Process Regression model is $551,332.65.
- MSE: The mean squared error is $330,802,649,442.17.
- RMSE: The root mean squared error is $575,154.46.
- MAPE: The mean absolute percentage error is 84.71%.

Interpretation

- The ARIMA model outperforms the other models in terms of MAE, MSE, RMSE, and MAPE, indicating better accuracy in predicting cab ride prices.
- The ETS model has higher errors compared to ARIMA, and Gaussian Process Regression exhibits the highest errors among the three models.
- Consideration should be given to the specific requirements of the application, as well as the importance of different types of errors (overestimation or underestimation) when selecting the most suitable model for forecasting cab ride prices.

SWOT Analysis & Conclusion

Implementing a SWOT analysis (Strengths, Weaknesses, Opportunities, Threats) for a cab ride forecasting project involves assessing internal and external factors that can impact the project's success. Here's a guideline on how you can perform a SWOT analysis for your project:

Strengths:

1. Data Availability:
   - The project leverages historical ride data from Uber and Lyft, providing a rich dataset for analysis and model training.

2. Diverse Modeling Techniques:
   - Implementation of various forecasting models, such as ARIMA, ETS, and Gaussian Process Regression, demonstrates flexibility and the ability to explore different approaches.

3. User-Centric Functionality:
   - The recommendation system is user-centric, considering factors like location, day of the week, and time, enhancing the overall user experience.

Weaknesses:

1. Model Accuracy:
   - The accuracy of the forecasting models, especially in terms of predicting cab ride prices, needs to be continuously evaluated and improved.

2. Limited Input Features:
   - The recommendation system currently relies on a limited set of input features. Expanding the feature set could potentially enhance accuracy.

Opportunities:

1. Real-Time Data Integration:
   - Integrating real-time data from cab services, traffic conditions, or events could improve the accuracy of predictions and recommendations.

2. Partnerships with Cab Companies:
   - Establishing partnerships with cab companies for access to more comprehensive and accurate data, potentially improving the model's performance.

Threats:

1. Competitive Landscape:
   - The project operates in a competitive landscape with potential new entrants or existing competitors improving their own forecasting systems.

2. Regulatory Changes:
   - Changes in regulations impacting the ride-sharing industry may affect the availability and quality of data, influencing the effectiveness of forecasting models.

In the evaluation of three distinct time-series forecasting models for cab ride pricing, the ARIMA model emerges as the most accurate and reliable. The ARIMA model exhibits lower errors across multiple metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). Its superior performance indicates a robust ability to capture temporal patterns and fluctuations in cab ride prices, making it a suitable choice for forecasting in this context. On the other hand, the Exponential Smoothing State Space Model (ETS) and Gaussian Process Regression, while providing reasonable forecasts, demonstrate higher errors compared to the ARIMA model. The recommendation system allows users to input their travel details and receive tailored suggestions for the most economical ride options based on historical data for the specified conditions. It enhances user experience by providing informed recommendations for selecting the most cost-effective transportation options between specified locations at a particular time.
The choice of the appropriate forecasting model should be guided by the specific requirements of the application and the importance of different types of errors in predicting cab ride prices accurately. Further refinements or optimizations may be considered to enhance the forecasting accuracy of the selected model.